New Algorithms for Evolving Robust Genetic Programming Solutions in Dynamic Environments with a Real World Case Study in Hedge Fund Stock Selection

Wei Yan

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of the
University College London.

Department of Computer Science
University College London

December 6, 2012

I, Wei Yan, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

This thesis presents three new genetic programming (GP) algorithms designed to enhance robustness of solutions evolved in highly dynamic environments and investigates the application of the new algorithms to financial time series analysis. The research is motivated by the following thesis question: what are viable strategies to enhance the robustness of GP individuals when the environment of a task being optimized or learned by a GP system is characterized by large, rapid, frequent and low-predictability changes?

The vast majority of existing techniques aim to track dynamics of optima in very simple dynamic environments. But the research area in improving robustness in dynamic environments characterized by large, frequent and unpredictable changes is not yet widely explored. The three new algorithms were designed specifically to evolve robust solutions in these environments.

The first algorithm 'behavioural diversity preservation' is a novel diversity preservation technique. The algorithm evolves more robust solutions by preserving population phenotypic diversity through the reduction of their behavioural intercorrelation and the promotion of individuals with unique behaviour.

The second algorithm 'multiple-scenario training' is a novel population training and evaluation technique. The algorithm evolves more robust solutions by training a population simultaneously across a set of pre-constructed environment scenarios and by using a 'consistency-adjusted' fitness measure to favour individuals performing well across the entire range of environment scenarios.

The third algorithm 'committee voting' is a novel 'final solution' selection technique. The algorithm enhances robustness by breaking away from 'best-of-run' tradition, creating a solution based on a majority-voting committee structure consisting of individuals evolved in a range of diverse environmental dynamics.

The thesis introduces a comprehensive real-world case application for the evaluation experiments. The case is a hedge fund stock selection application for a typical long-short market-neutral equity strategy in the Malaysian stock market. The underlying technology of the stock selection system is GP which assists to select stocks by exploiting the underlying nonlinear relationship between diverse ranges of influencing factors. The three proposed algorithms are all applied to this case study during evaluation.

The results of experiments based on the case study demonstrate that all three new algorithms overwhelmingly outperform canonical GP in two aspects of the robustness criteria and conclude they are viable strategies for improving robustness of GP individuals when the environment of a task being optimized or learned by a GP system is characterized by large, sudden, frequent and unpredictable changes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Genetic Programming (GP), pioneered by John Koza, is an important tool for solving problems automatically without being explicitly programmed. However, when GP operates in dynamic environments, particularly ones in which changes are frequent, large and unpredictable, GP-evolved solutions are often fragile and prone to failure [99, 16, 100, 123, 120, 18, 126]. This is the problem for which this thesis aims to offer viable solutions. Using the stock market as a case study, this thesis proposes three novel GP algorithms specifically designed to improve the robustness of solutions to challenging, dynamic environments characterised by frequent, large and unpredictable changes.

1. The first algorithm, 'behavioural diversity preservation', is a novel diversity preservation technique. The algorithm evolves robust solutions by preserving population phenotypic diversity through the reduction of population behavioural intercorrelation and the promotion of individuals with unique behaviour.

2. The second algorithm 'multiple-scenario training' GP is a novel population training and evaluation technique. The algorithm evolves robust solutions by training the population across a set of preconstructed environment scenarios, simultaneously, and by using a 'consistency-adjusted' fitness measure to favour individuals performing well across the entire range of scenarios.

3. The third algorithm 'committee voting' GP is a novel final solution selection technique. The algorithm enhances robustness by breaking away from 'best-of-run' tradition and creating a solution based on a majority-voting committee structure consisting of individuals evolved in a range of diverse environment dynamics.

1.1 Research Gap

GP has proven to be successful in a vast number of static applications [95, 132] and has been reported to achieve "human competitive"[96] results in many static applications including the evolutionary quantum computer programming [96]; the evolution of local search heuristics for SAT [96] ; the application of GP to the synthesis of complex kinematic mechanisms [160]; invention of optical lens systems [160]; the synthesis of interest point detectors for image analysis [160];

and GP evolved antenna flown on NASA's Space Technology 5 mission [109]. However, what happens if there are changes in the environment in which the GP operates? For a canonical GP, there is an implicit assumption of consistency in the evaluation function, because environmental information is accumulated by the system during successive iterations [121]. Unfortunately, the consistency assumption does not apply to many real-world engineering, economic and IT problems which require, in contrast, powerful heuristics that account for the uncertainty in the environment. Instances of problems in which the environment changes are abundant, for example designing modern aeroplanes that can maintain a flight path (direction, altitude and velocity of flight) against perturbations in atmospheric conditions, operating a chilled natural gas pipeline from Russia or Alaska under adverse suddenly changing temperamental climatic conditions, and even predicting US presidential elections .

This work (in Section 3.1) classifies dynamic environments into 4 categories, namely Type 1, Type 2, Type 3 and Type 4, with ascending order of degree of dynamism and, to facilitate explanation, a brief introduction of each type is given here. Type 1 environments are characterised by regular, explicit and predictable small changes; Type 2 environments are more challenging characterised by frequent, irregular and unknown changes; Type 3 environments are even more volatile characterised by continuous, highly irregular, highly unpredictable and more profound changes; and Type 4 environments are extreme environments characterised by abrupt and radical transformation changes.

A standard strategy for dealing with uncertainty and dynamism in learning problems is to regard each change as the arrival of a new problem instance that has to be solved from scratch (e.g. [88]). For Type 4 environments, this highly expensive approach often has to be adopted as the problem may alter completely after a change. However, this simple idea is often impractical in Type 2 and Type 3 environments for a number of reasons. For example, consider that the re-running of GP in dynamic environments can be performed either on a regular or an ad-hoc basis:

- If a GP system is to re-run regularly, its success depends closely on the choice of frequency. In a Type 1 dynamic environment where changes can occur regularly, it is relatively easy to determine an appropriate frequency after a close observation of the environmental changes. However, in Type 2 and Type 3 environments where changes occur randomly with a low degree of predictability, determining the frequency can be difficult or even impossible. If the system is re-run too frequently and the environment has not changed, effort is wasted. If the system is re-run too infrequently and the environment changes before the anticipated requirement for a re-run, the solution provided by GP is likely to be wrong, which can cause a disaster in some critical operations.

- If a GP system is to run on an ad-hoc basis whenever changes occur, its success depends on its ability to detect changes in the environment. With the exception of a Type 1 environment with regular changes, detecting changes can be problematic and as yet there is no effective detection method [166, 53, 81]. One problem is the difficulty in distinguishing real changes in the underlying environment from mere noise in the information, in other words over-detection. This causes a waste of resources due to re-running the system unnecessarily. Another problem is the difficulty in detecting changes in time. In Type 2 and Type 3 dynamics, large changes can happen very quickly without any warning. In such cases, even if changes are eventually detected, it is often too late to be useful.

Over the past few years, a number of authors (e.g.[166, 53, 81]) have proposed different strategies for dealing with dynamic environments; most of which could be classified as 'adaptive' methods designed for Type 1 environments. 'Adaptive' here refers to continuously tracking extrema or near extrema through the search space as closely as possible so that the resultant solutions are 'adaptive' [18] to a changing environment. On the other hand, the field of research on evolving robust GP solutions in Type 2 and 3 environments is yet to be largely explored [126]. The survey of the existing GP literature in Chapter 2 observes that there is very limited work focusing on robustness issues and of those reviewed none has so far dealt with Type 2 and 3 environments. This field opens up an important research gap and is the focus of this thesis.

Adaptive methods typically include multi-population(e.g. [20]), extra memory (e.g. [62]) and swarming (e.g. [28]). However, in Type 2 and Type 3 environments adaptive tracking does not perform well either because the environment changes too quickly, severely and frequently; or because it cannot be monitored closely enough; or because the changes occur after the commitment to a particular solution has been made. In such cases, in order to deal with large, frequent and unpredictable changes, creating *robust* GP solutions is a preferable strategy. This strategy has been widely observed across many species in nature against high level of uncertainties in their living habitats from the level of gene transcription to the level of systemic homeostasis [91, 90, 156]. Robustness and adaptive tracking, however, are not mutually exclusive; they can be combined if required. Robustness approaches would allow GP to continuously provide appropriate solutions to dynamic problems without the need for discontinuous operation or human intervention and can improve efficiency and efficacy in complex real-world domains.

1.2 Importance

Evolving *robust* GP solutions for Type 2 and Type 3 environments characterised by large, frequent and unpredictable changes is vital for the following reasons.

- Type 2 and 3 environments are extremely common in the real world given increasing uncertainty and volatility in natural systems and in the global political, social and economic systems. This work gives particular attention to economic systems, more specifically the stock market. The stock market is a fascinating example of Type 2 and 3 environments (very occasionally of Type 4). In the world of the stock market, risk, reward and catastrophe come in irregular cycles witnessed by every generation. Greed, hubris and systemic fluctuations have led to tulip mania in the 17th century [150]; the South Sea bubble in the 18th century [150]; the land looms in the 1920s and 1980s [149]; the U.S. stock market great crashes in 1929 and 1987 [149]; and the 2008 worldwide financial crises[125] to name just a few of well-known examples. But more importantly, it has increasingly become one of the most important economic forces, for example, in the last 60 years, the GDP share of the financial industry in US increased dramatically from 20% in the 40's to 80% in 2008.

- GP is an important automatic problem solving tool. GP is probably the closest development ever made towards Alan Turing's original "a machine that thinks": computers which automatically solve problems [162, 161], although it does have difficulties in evolving robust solutions in dynamic environments [30, 126] . It is especially suited as an automatic optimisation tool for complex dynamic problems given its unique advantages [126](detailed explanation in Section 2.1): it is a superior general-purpose search method [11]; it provides an explicit solution process breaking out of 'black-box' methods [54] ; its fitness functions make it possible to incorporate domain specific information [54]; and it is able to discover the underlying data-generation process of a series of observations using symbolic regression[30].

1.3 Thesis Null Hypothesis

When the environment of a task being optimised or learned by a GP system is characterised by large, sudden, frequent and low-predictability changes, the robustness of genetic individuals cannot be improved by means of the following algorithms.

Behavioural diversity preservation This preserves phenotypic diversity of a genetic population by reducing the degree of correlation between phenotypic behaviour of those individuals; greater phenotypic diversity leads to more robust solutions.

Multiple-scenario training This enhances robustness of solutions evolved by GP by means of training a population across a range of environment dynamics and selecting individuals that perform well in multiple scenarios.

Committee voting This enhances robustness of solutions evolved by GP by means of a committee structure whereby a small (odd) number of trained GP individuals offer solutions as votes, and the majority vote wins.

1.4 Contributions

1. The design and implementation of 3 novel GP algorithms for improving robustness of solutions.

    (a) 'behavioural diversity preservation' GP;

    (b) 'multiple-scenario training' GP;

    (c) 'committee-voting' GP.

2. Validation of the novel algorithms by simulating a real-world hedge fund investment strategy with real-world data.

3. Formulating a new robustness definition and metrics.

4. An empirical comparison and analysis of GP with a Support Vector Machine (SVM).

1.5 Publications

1. Evolving robust GP solutions for hedge fund stock selection in emerging markets, W.Yan and C.Clack, *Soft Computing - a fusion of foundations, methodologies and applications*, from the issue entitled "*Special issue on Bio-inspired Learning and Intelligent Systems*" Volume 15, Number 1, 37-50, DOI: 10.1007/s00500-009-0511-4, Springer Berlin / Heidelberg, 2010.

2. Behavioural GP Diversity for Adaptive Stock Selection, W.Yan and C.Clack, in Proceedings of the Genetic and Evolutionary Computation Conference (*GECCO 2009*), ACM. ISBN 978-1-60558-325-9/09/07.

3. Learning to optimize profits beats predicting returns - comparing techniques for financial portfolio optimisation, W.Yan, M.Sewell and C.Clack, in Proceedings of the Genetic and Evolutionary Computation Conference (*GECCO 2008*), pp1681-1688. ACM. ISBN 978-1-60558-131-6/08/07.

4. Evolving robust GP solutions for hedge fund stock selection in emerging markets, W.Yan and C.Clack, in Proceedings of the Genetic and Evolutionary Computation Conference (*GECCO 2007*), pp 2234-2241. ACM. ISBN 978-1-59593-697-4/07/0007. **Winner of Best Paper Award.**

5. Diverse committees vote for dependable profits, W.Yan and C.Clack, in Proceedings of the Genetic and Evolutionary Computation Conference (*GECCO 2007)*, pp 2226-2233. ACM. ISBN 978-1-59593-697-4/07/0007.

6. Behavioural GP Diversity for Dynamic Environments: an application in hedge fund investment, W.Yan and C.Clack, in Proceedings of the Genetic and Evolutionary Computation Conference (*GECCO 2006*), pp 1817-1824. ACM. ISBN 1-59593-186-4.

1.6 Thesis Structure

This introductory chapter mainly discussed the motivational background for designing new GP systems to evolve robust solutions. With the stage thus set, the remainder of this thesis is divided into six chapters:

Chapter 2 discusses the key characteristics, advantages and weaknesses of a canonical GP algorithm. It then considers previous work in dynamic environments, but it also reviews methods that have been proposed to categorise dynamic environments and describes the key financial terminologies in the context of this thesis.

Chapter 3 describes the design and implementation details of three novel GP algorithms, namely 'behavioural diversity preservation' GP, 'multiple-scenario training' GP and 'committee-voting' GP. It also offers a new robustness definition and a new categorisation of dynamic environments.

Chapter 4 presents the real-world environmental case study for evaluating the algorithms. The case study is concerned with selecting stocks for a hedge fund in an emerging market, Malaysia. It firstly highlights the important aspects of environmental characteristics of the chosen financial market, and then explains the inner mechanism, design and function of the fully automated hedge fund simulation system and the GP system. It also clarifies the data used and the experimental method.

Chapter 5 focuses on the evaluation of the experiments performed on each of the three novel algorithms in the context of the case study. It is divided into two parts: the first part on the experiment settings for each algorithm and the second on the analysis of experiment results for each algorithm.

Chapter 6 investigates empirically the qualitative system behaviour difference between GP and SVM. The details are provided on the experiment design and the analysis of empirical findings.

Chapter 7 concludes, reviews the contributions achieved and discusses future work.

Chapter 2

Background and Related Work

This chapter provides essential background knowledge on the key topics covered by the thesis. It has five sections. In the first section, the canonical GP model and its key advantages and weaknesses are explained in a nutshell; the second section reviews existing research on categorising different types of dynamic environments; the third section surveys prior works dealing with dynamic environments and the last two sections deal with the financial background, offering real-world examples of market dynamism and terminologies.

2.1 Canonical GP Model

This section explains the canonical GP model and highlights its advantages compared to many other intelligent systems. GP pioneered by John Koza enables computers to automatically solve problems without being explicitly programmed, and is one of the important emergent techniques in this broad area. In essence, GP is designed to solve program discovery [127]. The goals of program discovery are, according to Koza [94]:

"*Given a set of input-output pairs or formal specification of behaviour, produce a computer program that:*

1. *Non-trivially computes correct outputs for the inputs of each test case. Non-trivial computation implies that the program does not directly map from inputs to outputs by means of some sort of table. Rather, the program is an encoding of some algorithm.*

2. *Computes outputs in such a way that, if the inputs have been representatively chosen, it will compute correct outputs for novel inputs.*"

What makes a program-discovery algorithm useful is the fact that many problems from a wide range of domains can be translated into program discovery problems. Depending on the terminology of the particular field involved, the 'program' may be a game-playing strategy, an optimal design plan, a decision tree, an econometric model, or a stock selection strategy, perhaps more generally, a 'composition of functions' [94].

The canonical form of the GP algorithm as described in Koza's book [93] depicted in Figure 2.1 has the following steps:

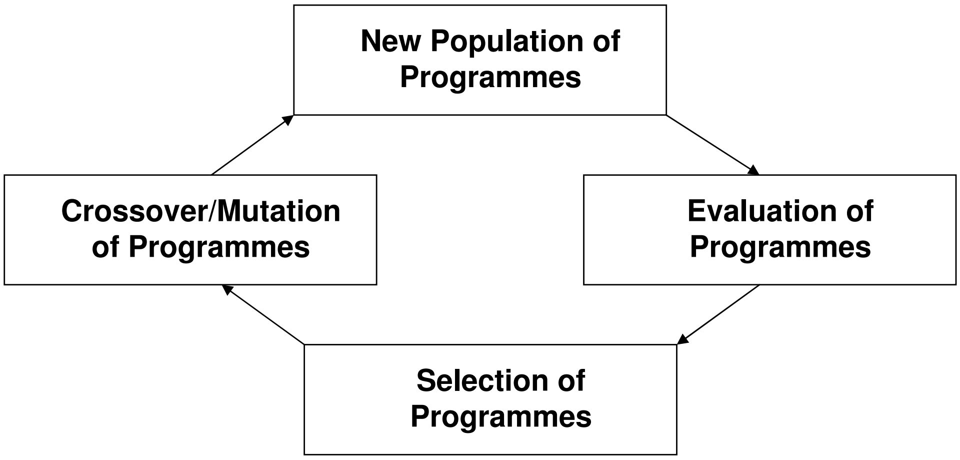


Figure 2.1: Canonical GP algorithm — evolution cycle

1. Generate an initial population of random compositions of the functions and terminals of the problem (computer programs).

2. Iteratively perform the following sub-steps until the termination criterion has been satisfied:

a) Execute each program in the population and assign it a fitness value according to how well it solves the problem.

b) Create a new population of computer programs by applying the following two primary operations. The operations are applied to computer program(s) in the population selected, with a probability based on fitness.

i. Copy existing computer programs to the new population.

ii. Create new computer programs by genetically recombining randomly chosen parts of two existing programs.

3. The best computer program that appears in any generation (i.e. the best individual so far) is designed as the result of GP. This result may be a solution (or an approximate solution) to the problem.

2.1.1 Key Components

To apply the standard GP to a given problem, there are five components to specify: 1) functions and terminals; 2) initial population; 3) fitness; 4) operators and 5) control parameters. Each of them is explained below:

Functions and terminals

In Koza's terminology [93], the representation of a GP program consists of terminals and functions. The size, shape and contents of these computer programs can dynamically change during the process. The set of all possible structures in GP is the set of all possible compositions of functions that can be composed recursively from the set of N functions from $F = f_1, f_2, ..., f_n$ and the set of N terminals from $T = a_1, a_2, \ldots, a_n$.

Initial structure

Each program individual can be expressed as a syntax tree. Figure 2.2 shows some examples of such programs expressed as trees. Koza [93] defined two methods termed full and grow. The full method works by selecting nodes from only the function set until a node is at a specified maximum depth when it then switches to selecting only terminals. Hence every branch reaches the maximum depth, creating more regular-shaped trees. The grow method randomly selects nodes from the function and terminal set which are added to a new individual, until a terminal node is reached, which terminates that branch.

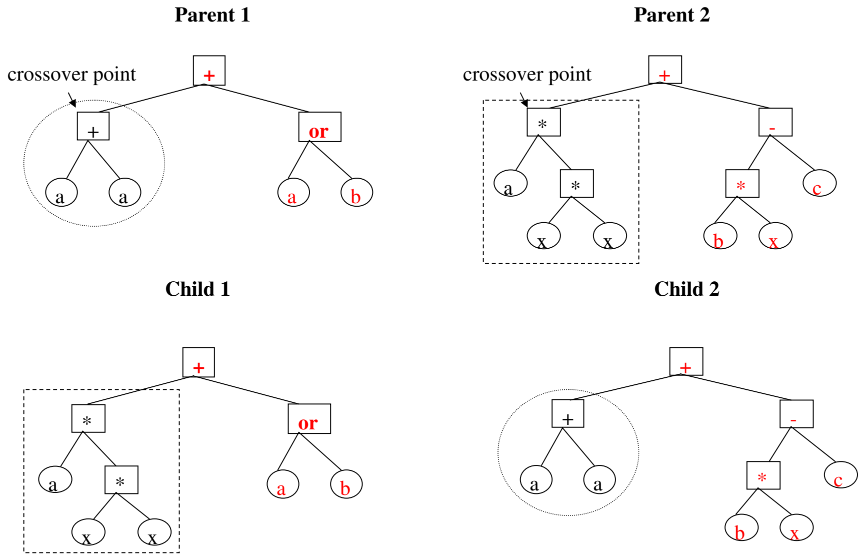


Figure 2.2: GP Trees and Crossover: Two crossover points are chosen and subtrees are exchanged to produce two children.

Fitness

Genetic programming requires an evaluation, or fitness, function to determine the quality of an individual. The fitness value is typically a single scalar value that allows selection methods to distinguish between various levels of individual quality. Multiobjective methods often use a vector to represent fitness. Most of the computation time, particularly for more complex problem domains, is spent on the fitness evaluation. To determine a fitness value, the individual is typically applied to a fitness case, or set of fitness cases. The score on the fitness case(s) is represented in the final fitness function value. The fitness can be expressed as the *raw fitness* or the *adjusted fitness*. The raw fitness describes the value that is assigned directly in relation to the problem. It is quite common for raw fitness to be described as a measure of error, with a small error representing a high fitness. The adjusted fitness represents the fitness of an individual as a value between 0 and 1, where 1 is the ideal fitness.

$$adjustedfitness = 1/(1 + rawfitness)$$

The "1+" is found in the denominator in order to avoid a division by zero situation. This will also have the effect of increasing the difference between two individuals who are at the higher end of the fitness scale.

Operators

GP shares many similarities with the broader domain of evolutionary algorithms, in particular in using crossover, mutation and selection operators in order to generate novel evolved structures.

- Crossover and mutation

  Crossover creates new offspring that consist of subtrees taken from each parent. It is a sexual operation in that there are two parents combining. Both parent individuals are selected from the population, a subtree in each tree is selected and the two subtrees are exchanged between the trees. For example, consider the two binary trees, Parent 1 and Parent 2, in Figure 2.2. Subtree selection is carried out by assigning a uniform probability to all internal nodes and leaf nodes separately. Then, an internal node selection probability, usually set to 0.9, defines the frequency of leaves or subtrees selected for crossover.

  Since in canonical GP all functions and terminals return and expect the same type, any exchange of subtrees between two trees will be valid. Mutation is often added alongside crossover in order to prevent premature convergence by introducing diversity back into the population. Subtree crossover tends to be the dominant operator in GP, while mutation operators are often used at lower rates.

- Selection operators

  The production and selection of new individuals is carried out in a *generational* or *steady-state* algorithm. In a generational algorithm, a new population is created from parents in the current population. In a steady-state algorithm, each new offspring replaces an individual in the existing population, either randomly or based on fitness.

  Fitness proportionate selection [93] is the most popular selection method that uses the adjusted or normalised fitness value of an individual to decide whether it can be a parent for crossover - the probability of selection is proportionate to its fitness value.

  Rank selection[93] is where each individual in the population is assigned a rank (from 1 to the size of the population, the least fit individual has the rank 1 and the fittest individual's rank is equivalent to the size of the population). The probability of each individual being selected for mating depends on its fitness normalised to the total fitness of the population.

  Tournament selection [88] does not require a centralised fitness comparison between all individuals. Instead, small groups of individuals ('tournaments') are compared for fitness. The fitter individuals in each tournament become parents for crossover with new offspring, replacing the least fit in each tournament.

Control parameters

GP uses various parameters to control a run. Such parameters include the population size, the number of generations and the depth of the parse tree that is used to represent the programs. Each of these should be chosen in a problem-specific way. A typical population might be around 400 - 500 to allow a large enough diversity in the initial set of individuals though many problems will require more than this. A number of generations might be around 50 - 100, though more generations may be required to arrive at a global optimum. The depth of the tree structure also depends on the complexity of the problem as well as the resources available, such as memory. There are no rules to describe which parameters should be used and often it is simply best to try different options and see which are most suited to the particular problem [126].

2.1.2 Analysis – Advantages and Weaknesses

Advantages

GP offers many unique advantages over all other approaches in machine learning or adaptive systems. These advantages can be summarised as follows:

- GP is a general-purpose search method and it achieves a better balance between exploitation and exploration than other search methods [11]. It is one of three main types of search methods defined in the current literature: calculus-based, enumerative, and random [65].

    1. Machine learning methods such as (Artificial Neural Networks(ANNs) and SVMs) are calculus-based methods. These methods either perform hillclimbing (by moving in a direction related to a local gradient) or set the gradient of the objective function to zero to solve a set of non-linear equations. They are local search methods. This constraint "severely restricts their application in many real-life problems, although they can be very efficient in a small class of unimodal problems." [10]

    2. Enumerative approaches, such as dynamic programming, evaluate every single point in a given search space. It is obvious that this approach is not suitable for complex problems with large or even infinite search space.

    3. GP is based on enumerative approaches but it does not employ a point-to-point search strategy, but instead uses a guided random search strategy. This strategy optimises the trade-off between exploration and exploitation [65]. A GP system explores a search space by spreading diverse search points across the space derived from a randomly generated initial population and variants resulting from crossover/mutation operations and at the same time it exploits the space towards peaks as selection disfavours inappropriate variants according to fitness criteria. GP is useful in problems where the search space is huge, multimodal and complex.

- The evolutionary based paradigm provides an alternative approach breaking out of 'black-box' methods [54]. Users have high degree of control over the entire process: for instance, users can explicitly and intuitively choose solution representations and fitness functions; solutions produced by evolutionary algorithms (EAs) often can be interpreted by humans;

users can observe the complete evolutionary loop from initial population creation to mating/combination process to fitness evaluation and to new population formation, making 'reverse engineering' (understanding how the algorithm learns and where the algorithm goes well or badly) a manageable task.

- Fitness functions in GP are not only concerned with statistical error measurements, but are more flexible in enabling the incorporation of problem-specific knowledge and issues – see Fogel [54]. Fogel further stated that "the problem of defining the payoff function lies at the heart of the success or failure: Inappropriate description of the performance index lead to generating the right answer for the wrong problem." For example, within many machine learning approaches or classic statistical methods, the quality of solutions is judged uniquely based on the squared error between the forecast and actual data. In contrast, within GP, the fitness function that is used to evaluate the appropriateness of alternative behaviour can be any definable payoff function without any restriction that the criteria be differentiable, smooth, or continuous [54] .

- GP is a knowledge discovery tool. It is able to discover the underlying data-generation process of a series of observations using symbolic regression to search for the right equation (or computer program) that produces the desired output for a given input in a given search space [30]. Symbolic regression differs from conventional regression types such as linear and polynomial regression, where the nature of the model is specified in advance by the user. However, a real-world problem, often quite reverse, is to decide what type of model most appropriately fits the data. Symbolic regression is particularly suited for this type of problem as it searches for both the function form and the appropriate numeric coefficients of a model.

Weaknesses

The genetic programming literature often states that the GP solution exhibits many difficulties in achieving robust performance in dynamic environments [99, 16, 100, 123, 120, 18].

Many researchers use the case of 'an artificial ant' (the Santa Fe trail) in order to draw some empirical insights into robustness of GP solutions. Langdon and Poli [103] observed that "GP performance is not dramatically better than random search", that it does not "guarantee a generalisation of the behaviour". Kushchu [100] observed that the GP test performance does not demonstrate robustness over minimal variations of the Santa Fe trail. When the 15 best individuals were tested on a rotated Santa Fe trail (no changes were introduced to the trail, but its mirror image was rotated by 90 degrees), the majority of individuals showed average, or below-average, performances. The artificial ant problem highlights the fragility of canonical GP in dealing with environment changes: solutions produced by GP can be 'brittle' and sensitive even to minor changes in complex environments. The ant problem also sheds light on characteristics of the environments that are hard for GP to deal with. To explain the performance of GP, it is necessary to consider the fitness landscape the GP sees. The fitness landscape of the artificial ant is characterised by a large, nonstatic search space and contains many false peaks, and many plateaus riven with deep valleys. This kind of environment bears a striking resemblance to environments of many real-world situations.

Various literatures in the field pinpointed four features of the canonical GP model which impede GP's ability to generate robust solutions. They are summarised below.

1. For problems in dynamic environments, GP gradually loses its capacity for exploration, reaching 'premature convergence'[103] because of the loss of phenotypic diversity in the population [73, 104, 59, 93, 137, 106, 32, 63]. The behavioural diversity preservation algorithms is designed to improve this feature (see Section 3.3.1 and 3.4.1).

2. The conventional training approach using a single environment scenario for dynamic environment does not provide sufficient learning opportunities for GP [100]. The multiple-scenario algorithm is designed to improve this feature (see Section 3.3.2 and 3.4.2).

3. The standard fitness evaluation method cannot assess the performance of an individual over several different environment cases [129]. Again, the multiple-scenario algorithm is designed to improve this feature (see Section 3.3.2 and 3.4.2).

4. A single 'best-of-run' solution model for dynamic environments is brittle, as the outcome has to rely on the accuracy of this single model [126, 129]. The committee voting algorithm is designed to improve this feature (see Section 3.3.3 and 3.4.3)

2.2 Related Works in Dynamic Environments

This section provides a survey of the techniques that have been proposed to make ECs suitable for dynamic environments and also a review of the existing EC applications in finance - stock selection in particular.

The existing techniques proposed to make ECs suitable for dynamic environments can be classified into two broad categories. The first category is 'adaptive techniques', referring to a range of algorithms which continuously track dynamic optima in a changing environment. The second is 'robustness techniques', referring to a range of algorithms that search for robust solutions to cope with changes in an environment. Figure 2.3 summarises related approaches discussed in this section.

2.2.1 Adaptive Techniques

Adaptive techniques attempt to track a global optimum as closely as possible in a changing environment, usually by constantly and adaptively optimising the location of the individuals in the search space so that the changes in the environment can be monitored and the global optimum tracked. These techniques make an important assumption that the kind of environments in which an evolutionary algorithm operates have only simple dynamics which can be closely monitored and tracked.

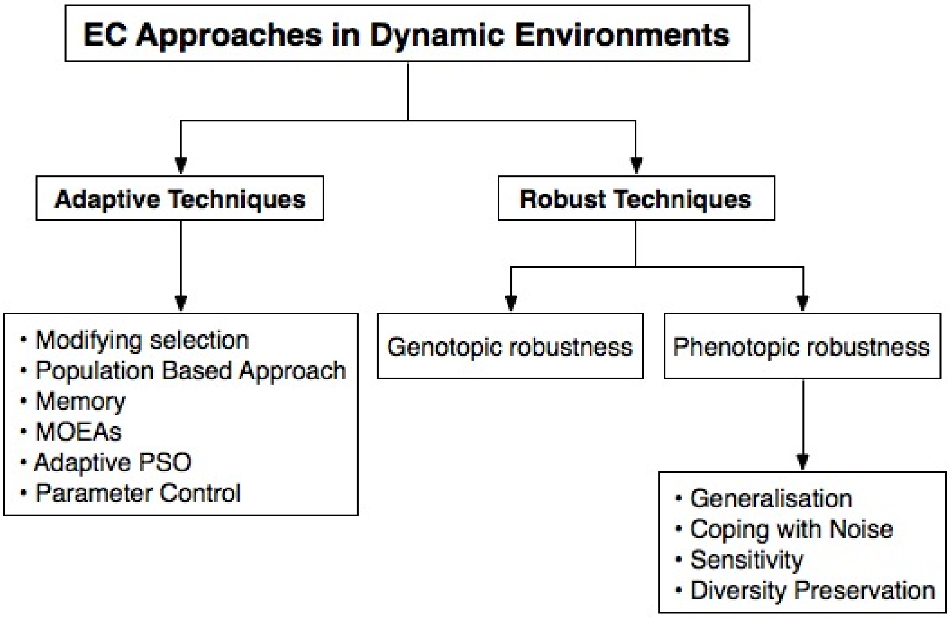


Figure 2.3: Categories of EC Approaches in Dynamic Environments

- **Population-based Approaches**

  These approaches employ two or more populations and attempt to spread the individuals across all possible peaks (optimum) in the landscape in order to increase the probability of tracking the correct global optimum. However, the population-based approaches share important drawbacks which limit their usefulness in challenging dynamic environments. Firstly, they are too complex to implement successfully, as users have to manipulate a large number of parameters which are often dealt with in an ad-hoc fashion. Secondly, and more importantly, such systems critically depend on the assumption that new 'peaks' or 'hills' are simply gradual movements of the old ones and thus can be tracked or monitored easily. However, such assumption cannot be made on many dynamic environments where new peaks may shift very quickly and/or occur in a distant region from old ones.

  - *Multi-population*

    One typical example is Self-organising scouts (SOS)[19, 21, 22, 20]. SOS allocates some 'child population' to peak regions of the search space while spreads out 'base population' in other areas. Progressively, all the peak regions should be covered by the child populations.

    This approach is to some extent similar to the multi-niches crowding design, as the search space is divided when a peak has been found. Results from experiments on simple and simulated problems are encouraging [19, 21, 22, 20]. However, the efficacy of this approach when applied to more complex problems or real-world problems remains to be explored.

- *Roaming Optimisation*

  The roaming technique uses an 'archive' external population (containing potential optima) to perform exploitation and subpopulation to perform exploration. [111].

- *Shifting balance*

  This technique [172] maintains the EA's exploratory power through evolving a number of colony populations separated from the core population. At regular intervals, the gene pool of the core population is diversified by receiving some emigrants from the colonies. [20] reported that this technique, when tested on a problem with frequent but small changes, is more efficient than a simple GA.

- *Multinational*

  The multinational was proposed by Ursem [163]. The main idea is to divide the population into nations or subpopulations, each corresponding to a potential peak in the fitness landscape. The subpopulations are grouped according to 'hills and valleys' in the landscape: for two points in the search space, the algorithm calculates the fitness of a number of random samples on the line between the points; a valley is detected if the fitness in a sample point is lower than the fitness of both end points.

• **Implicit or explicit memory**

Memory-enhanced techniques generally use redundant representations or extra memory for storing and retrieving good solutions at different stages of the evolution. However, efficacy of memory techniques is very restrictive because they are only suitable for small, regular and periodical changing environments where there are repeated occurrences of a situation [20]. Their performance in other types of dynamic environments remains problematic [183, 181, 22].

- *Redundant representations*

  Early works on redundant representations were not applied in the dynamic environment. Goldberg and Smith [62] report on experiments that use diploid and dominance. They employed a triallelic scheme where an allele can take on one of the three values '0', 'recessive 1', and 'dominant 1'. Kanta Vekaria (1998) implemented a GA 'selective crossover' mechanism similar to that of Goldberg and Smith but without diploid. It uses an extra vector, which accompanies the chromosome, to accumulate knowledge of what happened in previous generations, and uses this memory to bias and combine successful alleles (individual bits) during recombination onto the next generations. Ng and Wong proposed a new diploid scheme with four possible alleles (0 and 1, each dominant and recessive).

- *Artificial immune system (AIS)*

  The AISs are relatively new techniques for dynamic environments. Although they directly model immune systems, they do not offer fundamental innovations beyond existing techniques such as multi-population or niching and crowding. AISs have so far been used for detecting changes, not for dealing with changes. The main purpose of the immune system is to recognise dangerous foreign antigens (for example, bacteria, viruses) and the body's own cells or molecules. This particular quality of the natural immune system has inspired many researchers to apply AIS in anomaly detections. For example, Dasgupta [38] attempted to develop an efficient detection algorithm that could be used to notice any changes in steady-state characteristics of a system or a process. He assumed that the 'self' (that is, normal behaviour) exhibits stable patterns when observed over a time period. Any deviation exceeding allowed variation in the observed data is considered 'non-self'. His algorithm was successful in detecting the existing broken teeth in a cutting tool. Kim and Bentley [86, 87] used AIS for network intrusion detection. They focused on the 'dynamic clonal selection' approach in continuously changing network environments. Their approach essentially consisted of three populations of detectors, namely mature, immature and memory detectors. These detectors evolve in parallel, not only taking advantage of the previous experience of mature individuals, but dynamically generating new detectors whenever a new set of antigens is added into the system. The authors concluded that their algorithm was able to deal with a real environment where self behaviours change after a certain period and only a small subset of self antigens is visible at one time. [146, 182] proposed an immune system-based GA (ISGA) for dynamic environments. In their ISGA, there are two populations of individuals. The first population consists of plasma B-cell individuals and the individuals with the best match to the optimum (antigen) are selected and cloned. The second population is a collection of memory B-cell individuals which were the best ones at different stages of evolution.

- **Parameter Control**

  These methods can be classified into three categories: deterministic, adaptive and self-adaptive [45]. For example, many works have used various techniques to controlling population size during the evolution [158, 68, 61, 92, 148]. Because these techniques control one single parameter for an evolution run and ignore possible complex interaction between parameters, these techniques were found to offer a very limited improvement on an algorithm's adaptive tracking ability [181].

  1. *Deterministic*

     This takes place when the value of a parameter is controlled by some deterministic rules, i.e. a generation-based time-varying schedule, without using any feedback from the search [113].

  2. *Adaptive*

This takes place when the value of a parameter is altered using some form of feedback from the search. Usually credit assignment is used to determine the value of the parameter [147].

3. *Self-adaptive*

   The parameters to be adapted are parts of the the individual's the chromosomes. They can be regarded as extra genes encoded into the chromosome and they undergo mutation and recombination [8, 45].

- **Multiobjective Optimisation Evolutionary Algorithms (MOEAs)**

  MOEAs attempt to solve multiple and conflicting objectives and usually contain several optimal objective vectors representing different trade-offs among the objectives [186].

  There are two major limitations of MOEA techniques: 1) multiple objectives to be optimised by the algorithm have to be conflicting. This explains why some work (e.g. [101]) was not successful: the diversity promotion objective and their original problem objective are not conflicting but complementary; 2) it is very difficult to determine the quality of Pareto set approximations. In single-objective optimisation, quality can be defined by means of the objective function: the smaller (or larger) the value, the better the solution. However, in MOEA, often it is impossible to compare two solutions as neither dominates the other [187].

  MOEA calculates an individual's fitness on the basis of Pareto dominance [61]. Some approaches use the dominance rank [55] and others use the dominance depth [154, 41] or the dominance count (that is, the number of individuals dominated by a certain individual) [187, 188]. Investigation into the use of MOEAs in dynamic environments focuses on diversity preservation. In earlier works, fitness sharing is a popular technique adopted by MOGA [55], NSGA [154] and NPGA [70]. More recently, Bui et al. [101] added a diversity objective which is determined based on the Euclidean distance to solve the standard moving peak problem. They reported that the performance differences between the traditional GA and MO (multiobjective optimisation) approach were not significant, especially when random immigrants were introduced during the experiments.

- **Adaptive particle swarm optimisation (PSO)**

  PSO's population is composed of particles which each has a velocity and a location in the problem space. These particles converge to a global optima using memories to adjust their positions relative to a fitness goal at every generation. PSO approaches suffer similar shortfalls to the population-based techniques discussed in the previous section. PSOs critically depend upon successful detection of dynamic changes in the environment, which is difficult to achieve when the complexity of the environment increases, even by a small degree.

As with genetic approaches, the original PSO is designed for a static environment [85], and thus shares similar weaknesses with its genetic counterpart when facing dynamic environments. Carlisle and Dozier [28] made a first attempt at adapting PSO to dynamic environments by following the route of a 'restart' approach. The resetting is either done on a periodic or an ad hoc basis if the magnitude of environmental change is considered significant. Because it is a first look at dynamic PSO, their approach was tested on a couple of largely simplified cases and the results are inconclusive. Hu and Eberhart [74] introduced an automatic environment-sensing technique called the 'fixed-gBest-value' method. ('gBest' refers to the population global at which the best fitness so far has been achieved.) They believe that if the PSO cannot follow the dynamic changes, the gBest value will be trapped into the previous optimum value and thus never change. An optimum change is detected when the gBest value has not changed for a certain number of iterations. Parrott and Li [130] take inspiration from the GA-based multi-national approach which tracks multiple peaks and valleys in a dynamic environment (see previous section). The essence of their method lies in identifying appropriate particles and then grouping them into subpopulations. The key criteria is d - a measurement of Euclidean distance between two points in n dimensions. The model appears to be successful in a two-dimensional, three-peak space.

2.2.2 Robustness Techniques

Robustness is a very broad theme and it is impossible to capture all its aspects by means of a single definition. Robustness is considered to be a "ubiquitously observed property" and a "fundamental feature" of biological/evolvable systems [89]. Robustness is a property that allows a system to maintain its functionality despite internal and external perturbations [89, 168].

Although the need for producing systems with robustness has recently been recognised in the GP community, research dealing primarily with robustness is limited to only a few papers.

The definition of robustness in evolution systems varies from author to author but, in broad terms, it can divided into two categories:

1. **Robust with respect to internal changes (genotypic robustness)**

   - *Robustness as the resistance to changes from variation operators such as crossover and mutation*[153, 152].

     The authors observed that "the most outstanding evidence of pressure towards this type of robustness is the phenomenon of code growth (or code bloat) in GP". GP protects the useful codes by adding introns (code that does not contribute to the program's fitness [153]) so that they are not likely to be affected by crossover or mutation. The work of Piszcz [131] confirms that code growth is a requirement for GP robustness in complex solutions as it helps the evolved individuals to seek redundant building blocks to improve their resilience to crossover events.

- *Robustness as the ability to repair itself when subject to severe genotype damage* [118].

  This behaviour is reminiscent of autonomous regeneration of the pond organism hydra, which can reform itself when its cells are dissociated and then re-aggregated in a centrifuge [60].

2. **Robust with respect to external changes (phenotypic robustness)**

- *Robustness as the generalisation ability of the programs evolved* [98, 97, 15, 129, 120].

  The concept of generalisation originates from connectionist or symbolic learning research and is defined as "the desired successful performance of the solution when it is applied to an environment similar to the one for which it was evolved"[100]. In the context of evolutionary systems, the 'ability to generalise' is defined as "the predictive accuracy of the learner in mapping unseen input cases to outputs with a satisfactory degree of correction. [97]" In this respect, robustness is in line with, though opposite to, the definition of overfitting. Overfitting happens when the computational effort spent on obtaining a more precise fit to the sample results in an increased error in relation to other data. A formal definition is given by [119]: "*Given a hypothesis space H, a hypothesis h ∈ H is said to overfit the training data if there exists some alternative hypothesis h'∈ H such that h has smaller error than h' over the training examples, but h' has a smaller error than h over the entire distribution of instance.*"

  'Linear scaling' proposed by [83] is one of the most popular GP generalisation technique and experiments show some positive evidence of superiority using scaled error fitness measure over standard fitness measure for symbolic regression problems. However, recently, [34] critically examined generalisation performance of linear scaling technique and found that it does not generalise significantly better than the standard GP when tested on unseen data.

- *Robustness as the ability to cope with non-constant noise* [80, 124].

  Most practical problems involve noise. Some researchers [78, 134] [80] investigated this particular aspect of robustness when noise is added to the deterministic objective function values. These studies found that the use of noise can increase the "likelihood of robustness" [134]. However, these experiments do not specify the measure of robustness used and do not clarify whether the same level of robustness is achieved in an un-seen environment other than the traning environment. The efficacy of these methods remains unclear.

- *Robustness as the sensitivity of performance quality in the presence of external environmental perturbations.*

  This form of robustness is desirable because it is the most consistent with phenotypic robustness in nature. Although a biological system exhibits robustness in terms of genes and structures, et cetera, from an evolutionary perspective, ultimately robustness of only one feature matters: fitness - the ability to survive and reproduce (which in evolutionary systems means the performance quality of a solution).

[171] focused on cooperative coevolutionary algorithms, which are capable of exploiting the compositional nature of learning problems involving multiple, interacting agents. The algorithm is concerned with finding a team that performs well, but is also robust to deviations in individual member behaviour.

In Haynes and Wainwright [67], the environment where agents are evolved changes randomly at the end of each generation. However, the results have not been validated in out-of-sample data, the nature and the degree of robustness of the solutions remains uncertain.

A similar approach is taken by [120] for optimising manoeuvres in pursuer/evader problems in which new fitness cases are generated randomly at the end of each generation during evolution. The results are compared against the method using fixed fitness cases and it is shown that this approach reduces the brittleness of the solutions. However, there is a lack of a formal method that can be used to determine how the training and testing processes should be conducted, how fitness values are calculated and how the robustness should be measured. This is an aspect common to most of these experiments. In addition, this is a static problem and robustness is relatively easy to achieve using a simple technique.

- *Diversity Preservation as a robustness mechanism*

  This thesis considers diversity preservation as a robustness mechanism. [89] observed that for a natural system, robustness can be enhanced "if there are multiple means to achieve a specific function, because failure of one of them can be rescued by others". This mechanism encompasses diversity. Diversity, or heterogeneity, refers to the structure and/or behaviour differences between individuals in a population, "whereby a specific function can be attained by different means available in a population of heterogeneous components" [91]. For example, many organisms or species adopt diversity preservation strategies such as 'spreading of risk', 'escape in time and space' and 'bet-hedging' [170] in uncertain environments.

  In GP literature, diversity (or 'variety', a term used by Koza [93]) usually refers to genotypic diversity – the number of structurally unique individuals. Langdon [103] argued that genotypic diversity is "a sufficient upper bound of population diversity". However, [102] noted that measuring diversity with only unique genotypes "fails to consider the ancestral history of individuals, the degree of difference between non-unique individuals and their behavioural similarities".

  Numerous techniques have been developed to maintain population diversity. These approaches have three main research directions:

  (a) the preservation of genotype diversity based on formally-defined structural distance measures;

(b) the preservation of phenotype diversity based on the unique individual fitness in a population;

(c) the reintroduction of genetic material in various evolutionary phases.

Genotypic diversity preservation These approaches have been comprehensively reviewed by [65]. The main examples are

- 'Pygmie' algorithm [138] maintains diversity in terms of the length of an individual.
- Multiobjective optimisation [39] using three objectives ⟨fitness, size, diversity⟩. The diversity is measured based on edit distance between nodes.
- A structural difference measure [84] calculates edit distance between genotypes.
- A string edit distance [17] measures between fitness-contributing program codes.
- Diversity-guided evolutionary algorithm (DGEA) [164] uses 'distance-to-average-point' diversity measure flexibly during exploration and exploitation stages of the evolution.
- Fitness sharing with a tree distance diversity measure [47, 46] calculates the distance between every pair of individuals in a weighted arithmetic mean.

Phenotypic diversity preservation In the existing literature, phenotypic diversity is usually measured and preserved only at fitness value level. This kind of measure is sometimes called 'entropy[1]' [136]. High entropy means that there are many unique fitness values in a population and there is a low level of population convergence. Conversely, low entropy means that a population contains many individuals have the same fitness and there is a high level of population convergence.

McKay [116] and Hutter [76] work is an exception. It applied the fitness sharing concept [61] to test its feasibility in genetic programming. The phenotypic diversity was measured based on the number of fitness cases solved. Additionally, Mckay used negative correlation [108] and a root Quartic negative correlation [117] to preserve diversity.

Re-introduction One simple approach is just to increase the mutation rate. While using a high mutation rate can definitely increase the population diversity, it is at the cost of effective exploitation. In the later stages of a search, a high mutation rate tends to disrupt existing individuals that have well-coupled components. The resulting individuals usually have very low fitness and cannot survive at all. It is clear that diversity incurred by a high mutation rate does not imply an effective search. Similar reasoning also applies to the partial-reinitialisation approach, in which a portion of the converged population is replaced with randomly generated individuals. The

[1] Entropy is calculated for a population by first placing fitness values into classes and noting the size of each class.

No.	objective function	constraints
1	static	null
2	static	static
3	static	var
4	var	null
5	var	static
6	var	var

Table 2.1: T & M 6 classes of environments.1 and 2 are static environments; 3 to 6 are dynamic environments. *null*: the case where the set of constraints is empty; *static*: no changes in time; *var*: changes in time.

multi-restart approach is even worse. In this approach, when the population shows the signal of becoming trapped in local optima, an entire new epoch is started and the population is filled with the new random individuals [57]. This will waste all useful information collected in the previous epochs. The reintroduction techniques also have their limitations: 'random immigrants' and 'hypermutation' [56] do not perform well in continuously changing environments and abruptly changing environments, and 'Restart' techniques ignore the knowledge accumulated by the population from previous learning and are therefore expensive in terms of computing resources. None of the previously mentioned techniques are wholly satisfactory. Studies have shown [26, 9] that genotype diversity approaches "may not be useful for capturing the dynamics of a population" and phenotypic measures appear to have better run performance. However, phenotype diversity techniques do not explicitly consider behavioural diversity of individuals, although they successfully spread individuals across different fitness levels. In other words, by ignoring the behaviour of individuals, the diversity in the same fitness level is not maintained. Legg [106] remarked that "while the total population diversity was improved, the diversity among the fit individuals was not". Darwen and Yao [37] studied cooperation in the Iterated Prisoner's Dilemma problem and found that increasing behavioural diversity, not genetic diversity, can improve cooperation and performance. Using any one technique on its own appears not to preserve useful diversity, implying that more elaborate techniques should be explored.

2.3 Types of Changes in Dynamic Environments

In the real world, there are many different types of environments in which a problem can be encountered. However, categorisation of environments in the current literature is rarely dealt with and only four works in this area have been found so far.

The earliest work is by Trojanowski and Michalewicz (T & M) [159]. They categorise environments into 6 classes and each class is characterised by two variables, as illustrated in Table 2.1. The first variable is 'function landscape' and the second variable is 'problem constraints'.

The changes of function landscape take the following forms:

1. **random changes** – the next changes of the landscape do not depend on the previous change and the time t.

2. **non-random and non-predictable changes** – the next changes do depend on the previous changes, but the dependency is too complex to predict.

3. **non-random and predictable changes** – the change of the landscape is deterministic. These changes can be further divided into cyclical and non-cyclical changes.

4. **continuous changes**– changes are continuous in time.

5. **discrete changes** – changes appear in the environment from time to time and there are periods of stagnation between them.

K & M did not define the changes of problem constraints, the second variable. Instead an example of a manufacturing factory was given to illustrate what constraints are. In this example, constraints were a list of tasks, number of machines, repair time, et cetera. From this description, changes of problem constraints can be understood as objectives and resources available for a given problem. The constraints can be constant or discrete in time.

De Jong [40] described 4 categories of changes.

1. **Drifty landscapes** – The landscape undergoes small changes and the algorithm is able to track the optimum as it moves.

2. **Landscapes with morphological changes** – The landscape undergoes large and complex changes, for example new regions of fitness emerge where previously no optima resided.

3. **Cyclic changes** – This is where the changes happen periodically and regularly over time.

4. **Abrupt and discontinuous changes** – Sudden and cataclysmic changes.

Branke [19] developed another method of categorisation building upon the previous works. 4 main criteria were used by Branke.

1. **Frequency of change** – How often the environment changes.

2. **Severity of change** – This is usually characterised by the distance from the old to the new optimum.

3. **Predictability of change** – Are the changes purely random or do they follow a pattern that could be learned?

4. **Cycle length/ cycle accuracy** – This defines, for cycling environments, how long it takes until the environment returns to a previous state, and how accurately it will return to it.

More recently, [135] used two criteria which they believe to be the "two most prominent features of dynamic optimisation problems". The first criteria used is the *frequency of change*, as proposed by Branke. The second criteria is new, namely the *magnitude* which they defined as the relatedness of two success instances of a time-variant series or more specifically, the genotypic distance between successive global optima. Based on the definition, this criteria is in fact consistent with the *severity of change* proposed by Branke.

These works actually focused on categorisation of changes not of environments. A new categorisation focusing on environments is developed based on previous works, Branke's in particular, in the next chapter.

2.4 The Stock Market as an Example of a Dynamic Environment

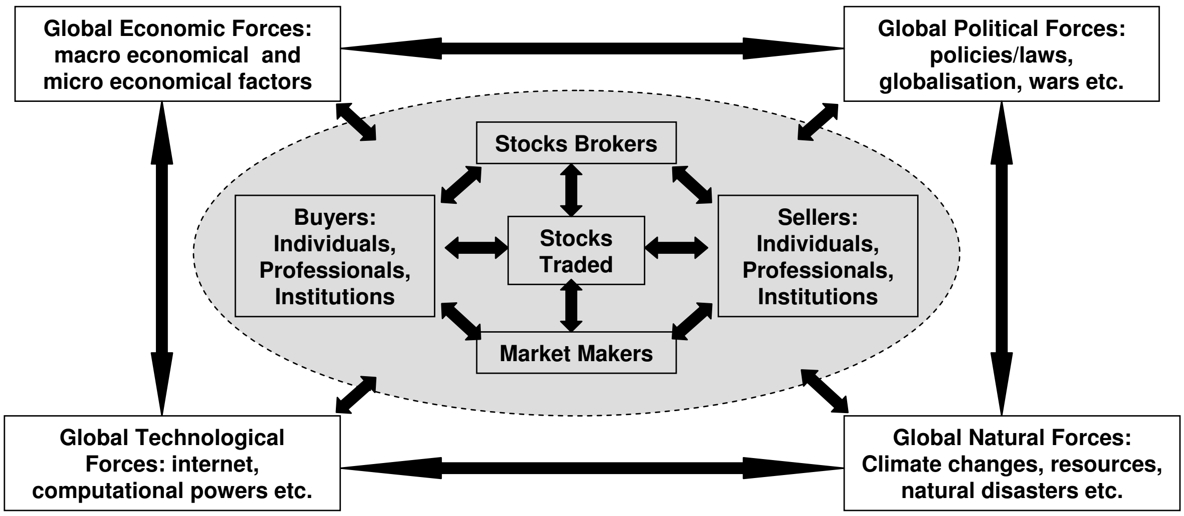


Figure 2.4: The stock market as a complex dynamic environment

The stock market is a fascinating example of a challenging dynamic environment. Figure 2.4 illustrates that the environment dynamics is influenced often simultaneously by both internal forces (different types of market participants) and external economic, political, technological and natural forces. These forces are ever changing and unpredictable and their interactions create the highly dynamic nature of the stock market. For example, according to a study by Economist [2], global economic crisis, capital mobility and IT boom have increased the importance of worldwide factors, in contrast to local country factors, in steering stock prices. The study found that the correlation between changes in American and European share prices has risen 100% since mid-1990s from 0.4 to 0.8. This dramatic increase means that "movements on Wall Street can now explain 80% of price movements in Europe".

Another example is that the current slower economic growth in the US has had negative

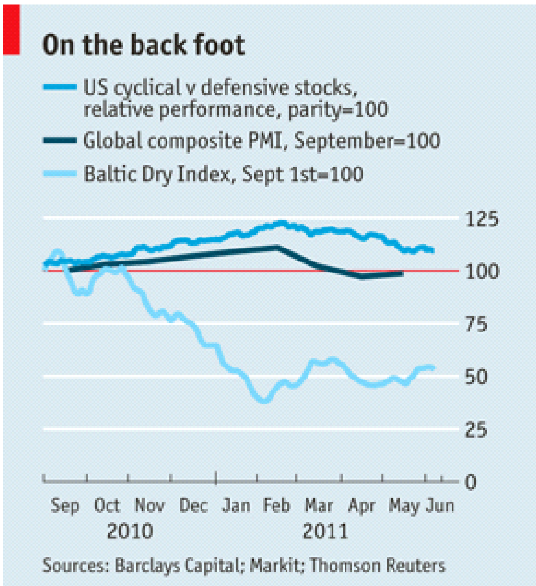


Figure 2.5: The equity markets are struggling in face of slower growth (Reproduced from [4])

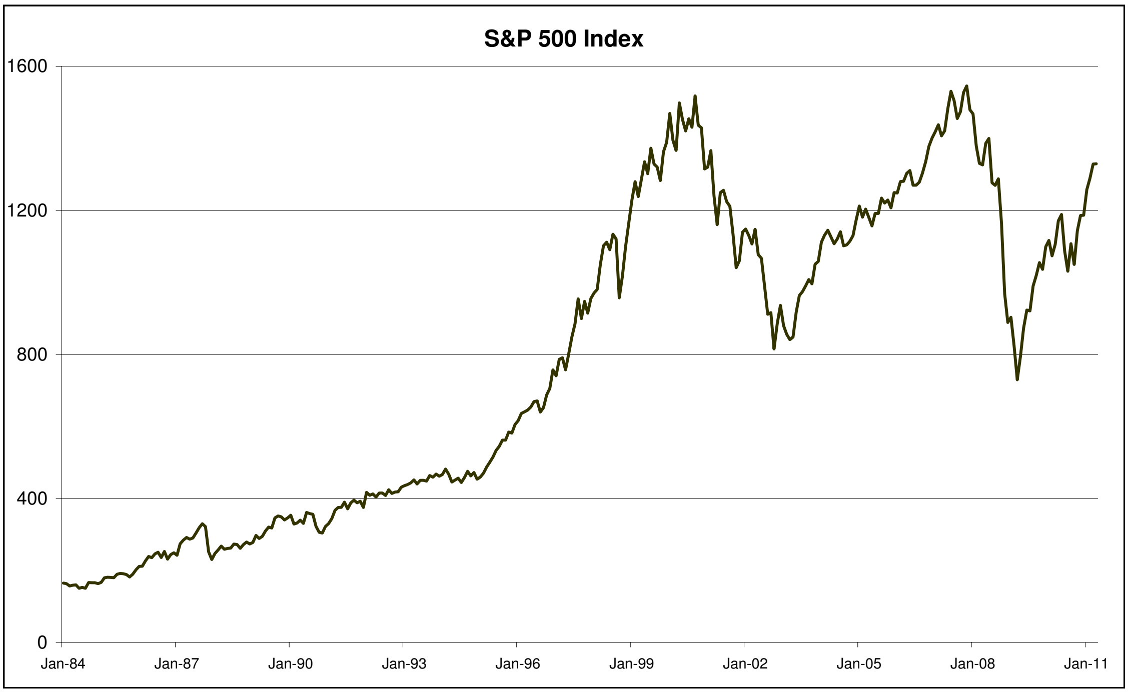


Figure 2.6: Stock Market Dynamics – S & P Index (data provided by Reuters)

impact on its equity market as illustrated in Figure 2.5 [4] . As the Baltic Dry Index, an indicator of global trading activity, started to turn down toward the end of 2010, the US defensive stocks

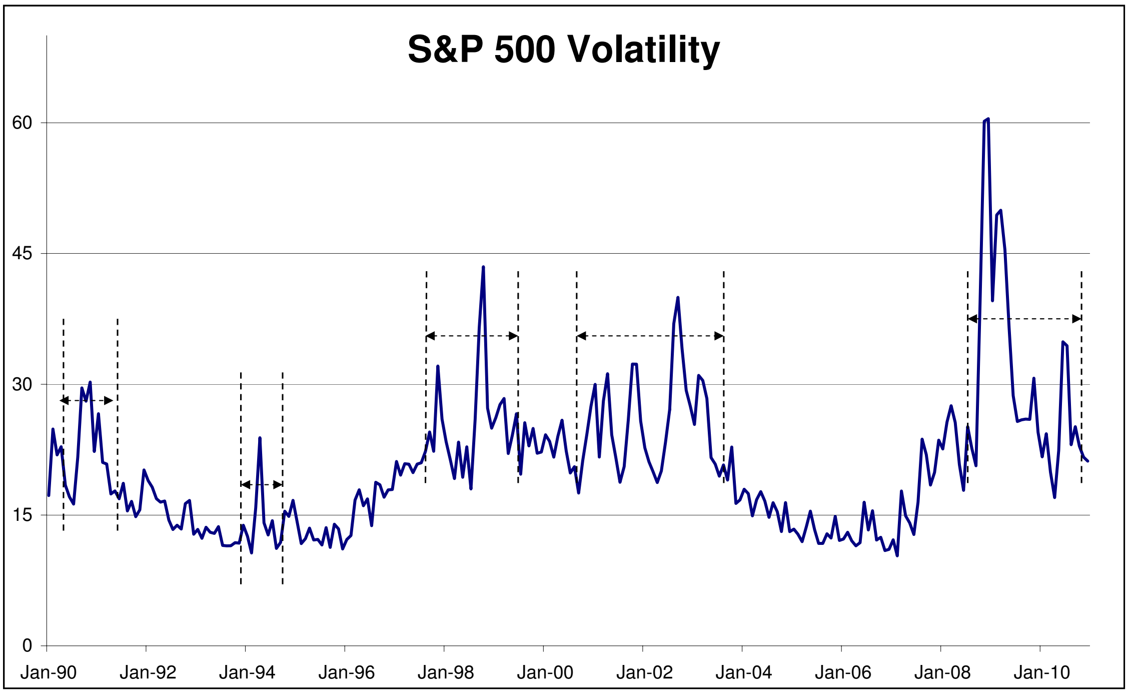


Figure 2.7: Stock Market Dynamics – S & P volatility (data provided by Reuters)

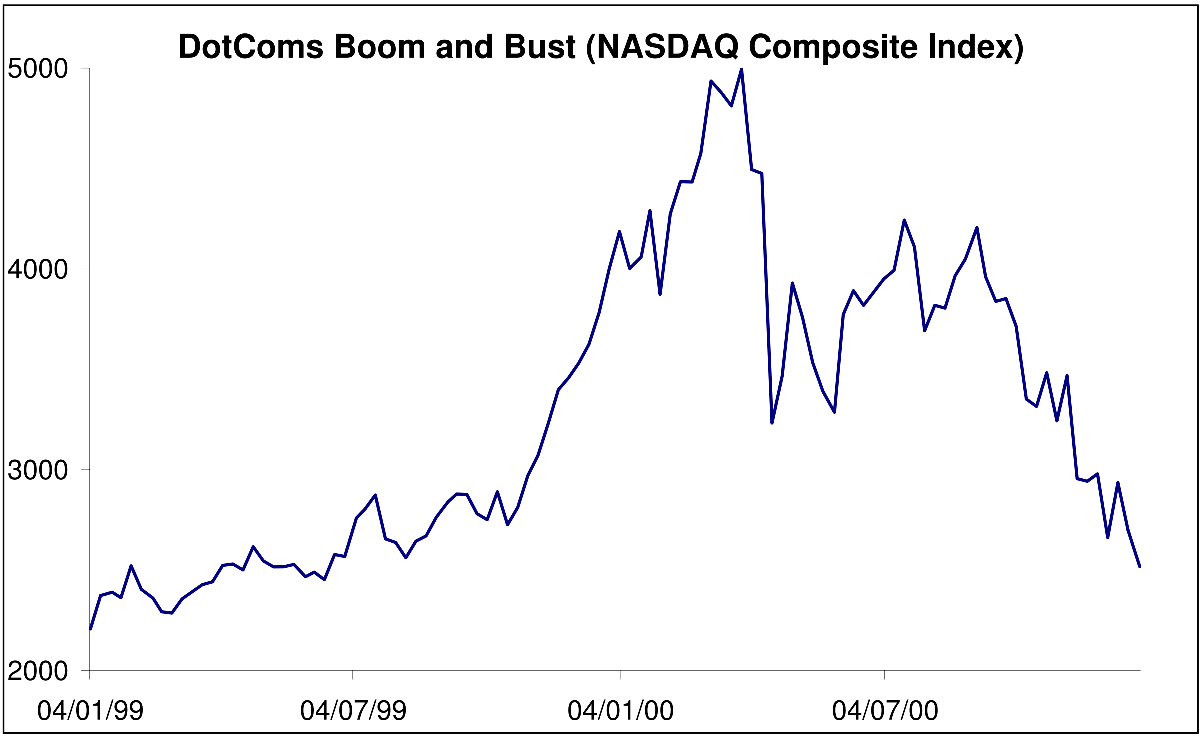


Figure 2.8: Stock Market Dynamics – DotComs Boom and Bust (data provided by Reuters)

including utility and food businesses which have relatively high level of consumer resilience outperformed the other businesses.

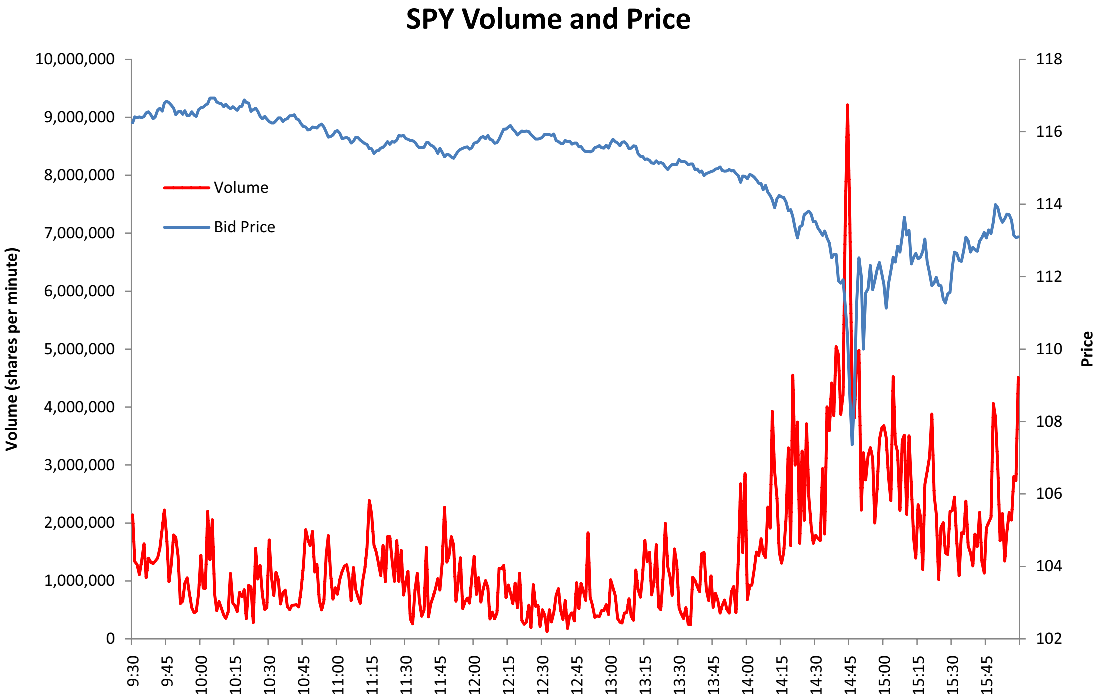


Figure 2.9: Black Swan : May 6, 2010 S & P 500 Market (Reproduced)

S & P 500 is one of the most important and traded indices in the world. It offers a good representation of dynamics of the US market. Figure 2.6 depicts S & P 500 index price movements and Figure 2.7 depicts S & P 500 index daily volatility. The figures very clearly show that the market undergoes irregular changes almost continuously. And the market often undergoes very large changes, as highlighted in Figure 2.7, and these changes often occur rapidly and are very difficult to detect or predict [1]. A good example is the boom and bust of DotComs during 1999 – 2001. Figure 2.8 depicts the extent of the market landscape changes before, during and after the boom. The rise of the e-commerce began in late 1999 and very swiftly caught the imagination of the investors. The value of NASDAQ increased three-fold in the period of 10 months, but very quickly the bubble went bust and NASDAQ lost more than third its value within a week. In such a volatile environment, few investors spotted the change when the bear market started with many billions lost and life ruined [3].

Very rarely, the market undergoes extreme and sudden changes caused by highly improbable/inconceivable dramatic events. These events are called the “black swans”[157]. The terrorist attack of September 11, 2001 is a typical example. All stock trading activities were suspended for a week in the States after the event and the market suddenly lost 10% (Dow Jones Industrial Average) when it re-opened and it took more than 3 months to recover. Another typical and very recent example is the May 6, 2010 volatility event as illustrated in Figure 2.9 which is reproduced from a report of US Securities and Exchange Commission [35]. According to the report, that afternoon, the prices of many U.S.-based equity products including indices in both the futures and securities markets “experienced an extraordinarily rapid decline and recovery” suddenly plummeting 5-6% in a matter of minutes before rebounding almost as quickly.

In summary, the stock market is a very challenging dynamic environment where changes are often large, irregular and difficult to detect or predict.

2.5 Stock Selection Strategies and Multi-factor Models

The goal of this section is to give essential background knowledge about stock investment in the context of this thesis.

The publication of Security Analysis by Benjamin Graham and David Dodd in 1934 started the era of modern money management. Graham and Dodd introduced the first systematic approach to stock-valuation based on company information. In the 1970s, a quarter-century of empirical work gave birth to two important theories of market price behaviour. The first theory was the Efficient Market Hypothesis (EMH)[50] which argued that the market is fully efficient and price changes follow a random walk and are thus unpredictable. The second was the Capital Asset Pricing Model (CAPM)[141] which argued that the stock returns are only associated with risk, in other words, higher returns imply higher risks and vice versa. By the 1990s, with the advancement of computing power and new statistical techniques for parsing data, these two theories has now been extensively revised by a new generation of empirical research. The empirical evidence shows that the stock market is largely inefficient and there lies substantial investment opportunity [79]; and CAPM is replaced by Multifactor models because stock returns are not necessarily related to risks but are also related by many other factors including microeconomical and macroeconomical factors [33, 79]. For more than a decade and a half, a new breed of investment strategies has emerged to exploit the market inefficiency. They specialise in market timing or value or emerging markets or macroeconomics, etc. Grouped together, these strategies are called hedge funds. Hedge funds have rapidly growing importance; for example, in the UK, they now account for more than half the daily turnover of shares on the London Stock Exchange [6].

2.5.1 Terminology

The financial terminology used in the thesis is explained in this section.

- **Long positions** — Stocks are brought by an investor who expects the prices of these stocks to go up between the purchase and resale period.

- **Short positions** — Stocks are sold by an investor, but the investor does not own the stocks which are borrowed from a third party, generally a broker, and he will return the stocks at a fixed future date. The investor expects the prices of these stocks to go down between the sale and return period.

- **Long-short strategy** — This is an active portfolio construction strategy that balances long positions in high-expected-return stocks and short positions in low-expected-return stocks. The strategy attempts to maximise portfolio profit and reduce the market exposure risk by exploiting long and short positions simultaneously.

- **Long-short return** — The main return of the portfolio derives from the return spread between aggregate long and aggregate short positions. Because the aggregate long positions and the aggregate short positions are of approximately equal value and have equal sensitivity to the underlying market, those portions of their returns that reflect overall market movements cancel each other out. All that remains is the return spread.

2.5.2 Related Work in Building Factor Models

Factor models capture the impact of various stimuli on returns, expressed either as linear or (more recently) non-linear combinations of the individual factors.

Traditionaly, financial models on the behaviour of stock returns typically use linear regression, the earliest example being the Efficient Frontier, established by Markowitz [115, 114]. Building on the Markowitz framework, Sharpe [141] developed what is now known as the Capital Asset Pricing Model (CAPM), a single factor model which specifies returns as a linear function of only systematic risk (beta). Three-Factor model by Fama and French [51] adds two other factors to CAPM, size and book-to-markt value. Arbitrage Pricing Theory (APT), a multi-factor model, extends CAPM and specifies returns as a linear function of various macro-economic factors.

In contrast, a number of recent studies have suggested the existence of non-linear structures in financial markets. Pesaran and Timmermann [66] found substantial evidence that stock returns follow a non-linear regressive process on various macro-economic variables such as interest rates and some non-linear terms such as the lagged values of the squared returns. LeBaron *et al.*[105] proposed a non-linear Adaptive Belief System (ABS) to capture the essential feature of a stock market where many different trading strategies compete and evolve. Following the same research line, Hommes[69] using market models based on non-linear adaptive evolutionary systems explained important stylized factors such as fat tails, clustered volatility, and long memory, of real financial series. However, discovering the optimal non-linear function in thousands financial variables remains an extremely challenging task and automated search algorithms such as GP can be a good candidate to undertake such a task.

Using GP to discover profitable trading models with technical rules has remained a very active research topic since mid-90's. These GP technical models, to some degree, are pseudo factor models, as each technical rule could be regarded as a factor. [122] and [112] adopted a GP approach to discover technical trading rules for the foreign exchange rates and stock market index (S&P 500). The results obtained by their research were encouraging and their line of research has been further explored by many researchers. For example, [42] developed a trading system based on a broad range of indicators for FX trading. Becker & Seshadri [12] used GP to evolve trading rules by simplifying representations used by Allen and Karjakainen for the S&P 500 index. [110] replicated Becker & Seshadri's work and observed that their approach provides better performance over buy-and-hold strategy than earlier attempts. More recent examples include that [48] used GP to generate risk-adjusted technical trading rules in Tehran Stock Market. [155] combined GA and GP. GA was used to optimise a weighted combination of technical rules. A similar design was also implemented with a GP that evolved complex rules from a pool of constituent rules and Boolean operators to combine them. [72] used GP evolved technical trading rules to investigate whether market capitalisation (size) is a significant factor to explain returns. [169] based on the work by [107] attempted to use GP to improve technical rule predictions on historical Dow Jones Average (DJIA) index data.

However, the relevant studies on evolving nonlinear factor models are very few in number and this research area remains largly unexplored. The closest example is by [13]. They used GP to evolve linear stock selection factor models to buy and sell stocks in S&P 500 universe. They reported that the portfolio returns were greatly enhanced using GP generated models comparing to the traditional CAPM and the three-factor model.

Past attempts to build non-linear factor models were predominantly focused on the use of the Support Vector Machines (SVMs).

Originally, SVM was developed to solve pattern recognition (classification) problems. However, with the introduction of Vapnik's ε-insensitive loss function [165], SVMs have been extended to solve nonlinear regression estimation problems and have been successfully applied to financial time-series forecasting [29, 128]. Although most of these works do not involve multi-factor models using only price data, there are a number of exceptions. For instance, SVMs were applied to forecast various global market index prices [31, 75] using basic macroeconomic variables as factors such as interest rates and GDP etc. These works found that SVMs outperformed ANNs. SVMs were also applied to forecasting exchange rates [27] using technical indicators as factors as well as stock selection [52] using a small number of accounting information as factors.

There are few existing studies on the comparison of regression SVM against EC techniques in finance. Bankruptcy detection is one of the popular areas; Vieira *et al* [167] concluded that GP achieved the best results for balanced datasets, but SVMs are more stable for unbalanced datasets. However, another study conducted by Alfaro-Cid *et al* [7] found that GP achieved very satisfactory results, improving on those obtained with the SVM using a highly unbalanced database. Zhang *et al* [185] compared credit-scoring models; they observed that although GP are better on average than SVM, the accuracy of SVM is more stable. Another area of interest is the prediction of insolvency in non-life insurance companies, where Salcedo-Sanz *et al* [139] found that SVM performed poorer than GP. The most relevant study probably is [25]. They compared GP and SVM in optimising a linear factor trading model and concluded that SVM produced better return with less risk than GP in predicting movements of Dow Jones Industrial Average (DJIA) index. This thesis has not found existing comparisons work on the topic of stock selection and portfolio optimisation. This is an important research gap and Chapter 6 attempts to make contribution in this area by offering empirical analysis on the differences between GP and SVM.

2.6 Summary

Chapter 2 summarised background information of the key topics covered by this thesis. It included a critical review of canonical GP, a survey of the state-of-art EC techniques in dynamic environments, a review of dynamic environment categorisation, an examination of stock market characteristics, definitions of financial terminologies and an overview of related works on automated methods for building factor models.

The first part presented some of the unique advantages of a GP system (in its canonical form) over other systems for solving many complex problems in dynamic environments. However, such a system has also exhibited many difficulties evolving robust solutions as observed by many researchers. This problem is mainly caused by four GP components spanning across the key building blocks of a complete genetic evolution cycle. They are described as follows:

- Training – The canonical training method cannot provide a sufficient level of training to GP in a complex, dynamic real-world environment.
- Evaluation – The canonical fitness evaluation mechanism cannot bias the search for a robust solution that can perform in all kinds of environments.
- Solution selection – While for static problems the best solution found during the run is typically implemented, this choice is not so clear in the setting of dynamic environments. A single best individual in terms of fitness during one run or one environment will not necessarily cope with different types of environments.
- Diversity – Canonical GP does not have an explicit diversity, especially phenotypic diversity, preservation mechanism which prevents premature convergence in optimised exploration of dynamic environments.

The chapter then investigated the past and current related EC techniques designed for dynamic environments. These techniques were critically reviewed under two different classes, namely 'adaptive techniques' and 'robust techniques'. Because the definition of robustness has varied considerably in prior work, an in-depth survey of many robustness definitions used in the domain of evolutionary computation was also presented. This is the first known comprehensive survey in this area.

From this comprehensive survey and analysis of different types of dynamic environment, two following research gaps associated with the existing works were identified:

- The adaptive approaches typically lack robustness and are not suitable for dynamic environments characterised by large, frequent and unpredictable changes. They tend to focus on the genotype or on cases in simple lab-controlled environments. This highlights the need for alternative approaches which focus on enhancing robustness of solutions for challenging dynamic environments.

- The existing definitions of robustness are vague and cannot be used for measuring robustness quantitatively.

It also described many categories of changes which could characterise dynamic environments. This highlighted not only the importance of taking into account the unique environmental characteristics in designing algorithms, but also No Free Lunch theorems [173] whose main gist is that algorithms cannot be universally good. The stock market is a very challenging environment, characterised by large, frequent and unpredictable changes. This chapter, using many historical real market examples, illustrated its unique characteristics.

The final part of the chapter briefly explained the historical development of stock selection strategies and factor models as well as some terminologies. It showed that market movements are determined by intricate interactions which can be expressed as non-linear multi-factor models between a web of different influential market elements. Given the high-level market complexity, building factor-models is a very difficult task and automated regression algorithms such as SMVs have been used for this purpose. However, GP applications were found to be very limited. It highlighted the need for comparative study in this area between evolutionary approaches and statistical approaches i.e. GP vs SVM.

The next chapter proposes three new algorithms which evolve robust GP solutions for real-world dynamic environments.

Chapter 3

New Algorithms

This chapter mainly focuses on design and implementation of the three new GP algorithms designed specifically to improve robustness of solutions and is primarily based on our published work [175, 177, 174, 179, 180]. The first two sections of the chapter explains our own method of categorisation of dynamic environments and a new definition of robustness. These two parts are indispensable for the following two reasons: 1. Different types of environmental dynamics require different approaches that take into account the unique environmental characteristics associated with each type and one approach cannot be suitable for all types of dynamics (there is 'no free lunch'); 2. The new definition of robustness provides quantitative measures which are critical for validating new algorithms.

3.1 Categorisation of Environmental Dynamics

Categorisation of environments is important because different types of dynamics require different approaches that take into account the unique environmental characteristics associated with each type. As illustrated in Table 3.1, this thesis classifies dynamic environments into four types based on the four environmental characteristics identified by the previous works, Branke [19] and De Jong [40] in particular, reviewed in Section 2.2. The four characteristics are: frequency of change, predictability of change, landscape morphological change and visibility of change.

In Type 1 environments, the changes take place rarely, regularly and explicitly with high degree of predictability and the fitness peak drifts slowly and slightly from one region to another nearby. Type 2 environments display higher level of dynamics than Type 1 with higher frequency change, lower predictability and visibility, and the landscape undergoes more important changes. Type 3 environments have an even more complex level of dynamics than Type 2. In these environments, changes are continuous over time, highly irregular and difficult to predict. Although the fitness landscape before and after a change display some reasonable degree of exploitable similarities, their morphologies undergo profound changes. Type 4 have the highest level of dynamics. In these extreme environments, abrupt, unpredictable, undetectable changes make fitness landscape transform entirely without any reference to the history. Complex real-world dynamic environments such as stock markets may exhibit Type 2, Type 3 and Type 4 characteristics over a period of time. But Type 4 changes are extremely rare, they are so called

environmental characteristics	**Type 1**	**Type 2**	**Type 3**	**Type 4**
frequency of change	low	medium to high	continuous changes	continuous changes coupled with abrupt and discontinuous changes
predictability of change	high predictability with regular pattern or trend in the change and completely non-random	medium predictability with some detectable patterns	low predictability, difficult to predict direction, time, or severity of the next change	completely random changes and impossible to predict direction, time, or severity of the next change
landscape morphological change	landscape undergoes very slight changes and new optima very close to previous ones	landscape changes morphologically with current peaks being destroyed and new and remote peaks arising from valleys, but new optima are trackable	landscape undergoes large and complex changes, but does not change completely with some reasonable degree of exploitable similarities	landscape changes completely with no exploitable similarities
visibility of change	occurrences of a change explicitly known to the system	occurrences of a change sometimes known to the system	occurrences of a change not known to the system	occurrences of a change not known to the system

Table 3.1: Types of Dynamic Environments

'black swans' described in Section 2.4. This work focus on Type 2 and Type 3 environments which are highly representative in these environments.

3.2 New Robustness Definitions

The purpose of redefining robustness is to highlight specific aspects of robustness that are believed to be fundamental in the context of dynamic environments and to facilitate the development of quantitative measures for robustness.

Solutions *S1* and *S2* are trained in an environment T and used in a previously unseen environment V (which is different to T).

In environment V, their performance is measured at discrete time intervals. The key terms for determining robustness are

1. overall fitness scores

2. the standard deviation of fitness scores measured at each time interval (stability of fitness)

3. the variation in overall performance between the training environment T and the unseen environment V (percentage rate of fitness change)

A solution *S1* is defined to be more robust than a solution *S2* if any of the following apply:

1. *S1* and *S2* have similar standard deviations, and *S1* has a higher overall fitness than *S2* in V.

2. *S1* and *S2* have a similar overall fitness, and the standard deviation of S1 in V is smaller than the standard deviation of *S2* in V.

3. *S1* has smaller percentage rate of overall fitness drop from environment T to environment V than *S2*.

Robustness of solutions evolved by proposed new algorithm will be evaluated in accordance to this new definition later in the thesis.

3.3 New Algorithms

Three new GP algorithms are proposed in order to enhance robustness. The three new algorithms are:

- *Behavioural diversity preservation.* For Type 2 dynamic environments, it enhances robustness of GP solutions by means of preserving phenotypic behavioural diversity of a genetic population through the reduction of degree of correlation of behavioural patterns among the GP individuals.

- *Multiple-scenario.* For Type 3 dynamic environments, it enhances robustness of GP solutions by means of training the whole population across a range of environmental scenario dynamics and selecting individuals that perform well in multiple scenarios.

- *Committee voting.* For Type 3 dynamic environments, it enhances robustness of GP solutions by means of using a committee structure whereby a small (odd) number of trained GP individuals offer solutions as votes, and the majority vote wins.

All of the three algorithms are substantially different to prior work in this area. This section firstly highlights the unique features of each algorithm, including the design logic behind them, and Section 3.4 provides a detailed explanation on their implementation.

3.3.1 Behavioural diversity preservation

This algorithm takes its inspiration from the notion of 'diversification' in financial management, which creates portfolios that are more robust to changes in the market environment. [1] The notion of not "putting all one's eggs in the same basket" has always been a tenet of commonsense financial wisdom. Modern Portfolio Theory (MPT) (as developed by pioneers H. Markowitz and W. Sharpe [142]) formalised this conventional principle. Before MPT, a naive portfolio manager may have believed that s/he was conservatively lowering the risk profile of their portfolio by spreading their assets over 100 shares rather than over 10 shares. MPT showed that this was not necessarily the case. In fact, a portfolio of 10 shares made of stocks with low intercorrelations could be much more 'efficient' than a portfolio of 100 narrowly-correlated stocks.

To be able to use the power of statistical concepts, MPT developed a formalised representation of financial instruments. Each instrument is regarded as a random variable, yielding returns which follow a probability distribution. As a first approximation, all probability distributions are assumed to be normal. Based on that representation, the 'risk' of every financial instrument is represented by (a function of) the standard deviation of the distributed returns. What then is the best way to go about selecting a portfolio of stocks? Within the framework of MPT, a portfolio is a linear combination of financial instruments and each portfolio is itself a 'return variable', yielding returns with a mean equal to the linear combination of the mean returns of each instrument included in the portfolio, and with standard deviation (risk) depending on the variance and covariance of the financial instruments included in the portfolio. The desirable or 'efficient' portfolios are portfolios endowed such that it is impossible to find another portfolio which exhibits a higher mean return without having also a higher standard deviation (risk). By analogy, it is proposed to observe a principle of diversification in selecting the individuals allowed to survive and recombine in the new GP algorithm (NGP). Whereas canonical GP selects only high fitness individuals for survival, individuals are treated as financial instruments and their intercorrelation is looked at. The total population is then stratified into fitness subgroups. For each subgroup, a low level of intercorrelation is constantly maintained so that the individuals permitted to survive and recombine present behavioural diversification. This algorithm differs from other 'layered' approaches (the population are classified either based on fitness or age)[71] by inserting an additional diversity preservation mechanism which monitors intercorrelation between individuals.

Figure 3.1 shows that three new mechanisms (highlighted in grey) are used to preserve both fitness diversity and behavioural diversity.

- **Multiple fitness segments**

[1] Because the algorithm is inspired by the key financial notion and then later in the thesis it is evaluated in the context of financial portfolio management, the algorithm may be misunderstood to be only suitable for specific financial applications. It is to be stressed that the key features of the algorithm demonstrate that it is a general algorithm and that it can be applied to any time-series related problem domain.

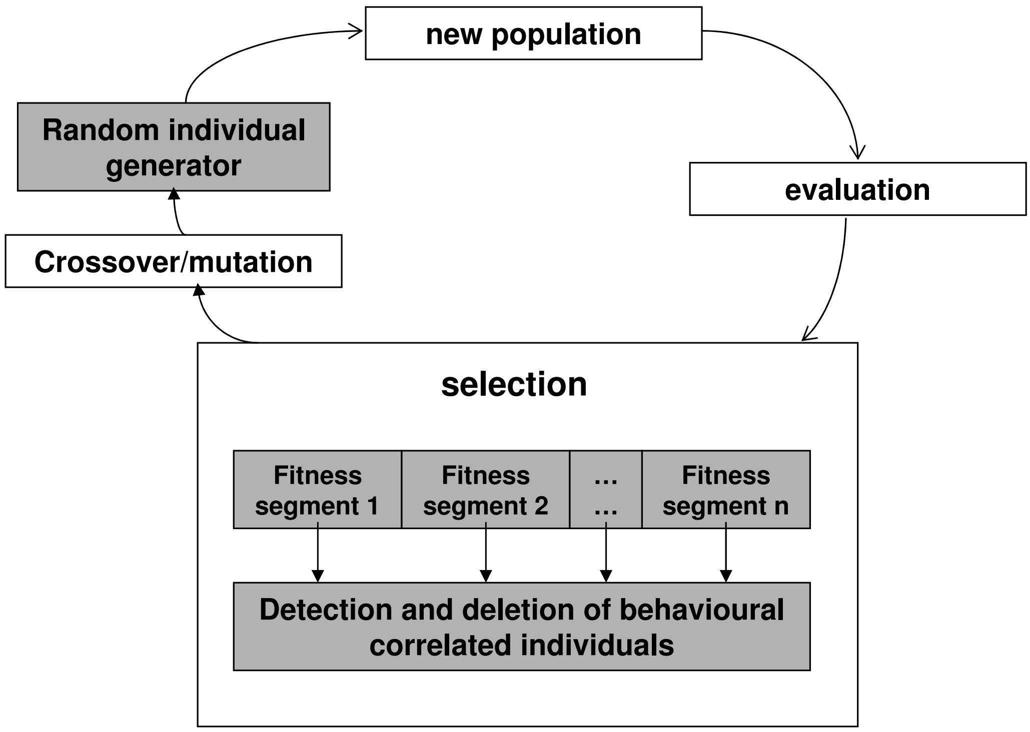


Figure 3.1: Behavioural Diversity Preservation GP

Multiple fitness segments are used to classify individuals based on their fitness value. These segments accommodate individuals within a specified range of fitness, and the entire range of possible fitness is spanned by the union of the fitness range of all segments. Each segment has a lower-bound fitness threshold and upper-bound fitness threshold determined adaptively.

This organisation of the population extends the search in the search space. It can also be understood as a way to maintain many diversified phenotypic groups, using fitness as an indication of an individual's phenotypic diversity within the *global* population. Unlike canonical GP, individuals on relatively unpopulated fitness segments are given an opportunity to survive, and are allowed to mature continuously.

- **Behavioural diversity measure**

  This measure is designed to monitor 'behavioural diversity' in the population. The term 'behavioural' instead of 'phenotypic' is used here because of two reasons.

  - The first of course is that it emphasises the distinction of this diversity approach from genotypic approaches (which aim to preserve GP tree structure diversity) taken by many existing work (details see Chapter 2).
  - The second is that it also emphasises its difference from existing phenotypic ap-

proaches which typically preserve fitness score/entropy level diversity (details see Section 2.2.2). The new algorithm aims to preserve behavioural level diversity, an extra dimension beyond fitness. The fitness score generally indicates the survival ability. But the behavioural dimension goes deeper, it indicates the observable characteristics or traits of an individual in a given environment. The relationship between the fitness dimension and the behavioural dimension is one-to-many which means that two individuals with the same fitness score can exhibit very different behavioural traits. For example, consider two GP individuals representing two trading strategies X and Y having the same track record of total profitability (fitness) of 10% over the past 4 years with year 1 to year 4 profitability -20%, +10%, -20% , 40% and +3%, +2%, 2.5% 2.5%, respectively. In this example, if one looks beyond fitness, X and Y clearly have different phenotypic behaviour, in other words, there is a low degree of behavioural correlation between X and Y: X is volatile with big losses and gains; in contrast, Y is consistent with modest gains. This kind of approach is more coherent with the definition of phenotypic behaviour in biology.

The logic behind preserving behavioural diversity is straightforward. Consider an individual that has a behaviour (the way it reacts to the environment) that is very different from other individuals at the same fitness level: if that individual were to be deleted, this would decrease the total population diversity of phenotypic behaviour. Conversely, if a set of individuals has correlated behaviour then all but one from this set can be deleted without comprising behavioural diversity.

The behavioural diversity measure is needed to monitor constantly diversity at behavioural level. During the evolution process, the phenotypic behaviour of each individual within the same fitness segment is recorded along with the phenotypic behaviour of the fitness segment it belongs to. Then measurements are taken of the degree of behavioural correlation between the individual and its fitness segment. If the difference of behaviour pattern between an individual and its segment is large, then it means that this individual exhibits either largely different behaviour from other individuals, or unique behaviour which one would strive to preserve. Individuals found to have highly correlated behaviour are chosen for deletion from the fitness segment.

- **Random individual generator**

  A random individual generator feeds new genetic material (in the form of individuals) continuously into the bottom processing level . It is beneficial if this generator is made to be unbiased such that it is able to supply a complete set of all possible low-level building blocks, unless there is prior knowledge about the search space that should be used to bias the generator. Of course, new building blocks may be discovered by recombination and other genetic operations and this unbiased conditioning is not strictly necessary, however, the inflow of random individuals relieves NGP from depending on a large population size to provide and preserve enough different generic material at the outset.

3.3.2 Multiple-Scenario approach

This algorithm is designed to improve GP solution robustness in Type 3 extreme environments by using multiple scenarios in training and fitness evaluation favouring individuals that perform well across multiple environment scenarios. During the GP runs, a well-structured training process is particularly important (or even a prerequisite) for maintaining robustness in highly dynamic environments.

The multiple-scenario training approach is inspired by the widespread Case-Based Learning (CBL) in the fields of medicine, law and business, and Case-Based Reasoning (CBR) in machine learning. The case-based methodology makes learning efficient and effective by exploiting past experience in similar context, and by creating wide exposure to a variety of case dynamics, so that knowledge about how to deal with various difficult and extreme environment situations can be learnt and/or experienced during the training. The methodology also overcomes the sampling difficulties in complex dynamic environments - through selective sampling of the environment and avoidance of sampling excessively large samples, which is often practically unfeasible and inefficient. In this algorithm, during the training, the GP population is simultaneously fully exposed to a variety of distinctive environment scenarios, especially difficult and extreme scenarios. There are 3 unique features in this algorithm as depicted (in grey) in Figure 3.2.

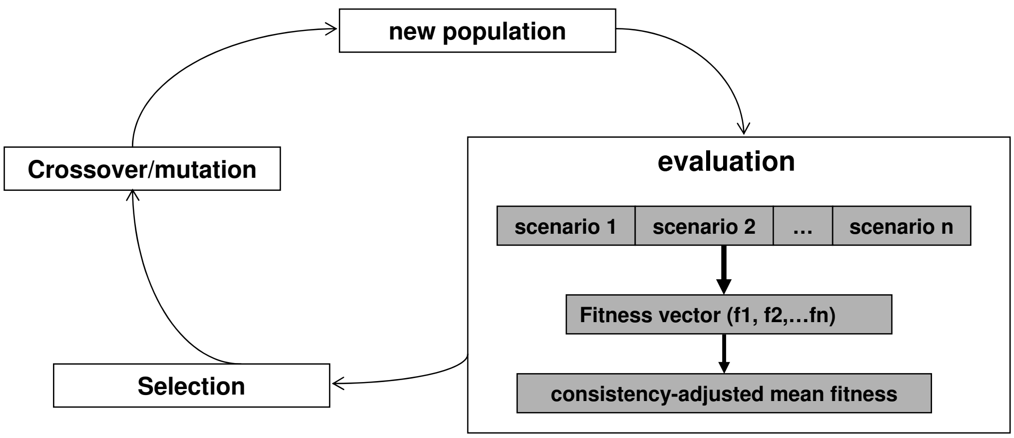


Figure 3.2: Multiple-Scenario Training GP

- **Construction of environment scenarios**

  The environment scenarios are constructed in two phases. During the first phase, the assortment of the environment scenarios for training is determined. Those scenarios represent either classical environmental characteristics or broad environmental trends. During the second phase, a large time-series data is broken down into a number of series of sub-component data, according to predetermined characteristics and trends, for constructing scenarios. These series of sub-components become the multiple scenarios for training GP individuals. These two phases could be achieved either by user manually or by computer automatically. In this thesis, the scenarios are constructed manually because the relatively small data size and small number of scenarios make the task completely manageable. Of course, the automatic approach is preferable for better integration with the main system and errors reduction when data is large and scenarios are complex.

- **Training**

  Multiple-scenario training can be integrated into the GP system at two different stages.

  - The set of training scenarios is integrated with the whole evolutionary process, each individual in the population is evaluated repeatedly under the various scenario settings during generational selection process. This is probably the most natural approach.
  - The set of training scenarios is only applied to the individuals in the final generation, by re-evaluating each individual in the final population several times. If one considers the individuals in the last population as representative of the knowledge gathered by GP, it is reasonable to simply re-evaluate individuals in the final population. This option also has the advantage, in terms of minimising computational cost, of fewer evaluation iterations.

- **'Consistency-Adjusted' mean fitness**

  A new measure of fitness is defined for use in dynamic environments. Since the analysis is concerned with the performance of the GP individuals across the entire range of environment dynamics, the unit of measure shall be considered to be: the entire fitness trajectory collected across GP exposure to a large sample of the scenarios. To begin, the mean fitness is defined as the simple average of fitness values over all scenarios. The definition is similar to the 'on-line performance measure' originally suggested by DeJong[40]. The individuals are also measured in terms of overall performance consistency which serves as an indication of the method's ability to cope with large environmental changes (robustness dimension 2). Statistical variance is used to measure the effect of fitness value fluctuations. The two measures are combined in a single formula to produce a single value which is designed to provide an aggregate picture of an individual's performance, where the performance information has been collected over a representative scenario of environment dynamics.

3.3.3 Committee voting

The committee voting algorithm is designed to enhance the robustness of the GP solution in a type 3 dynamic environment by creating a committee-based final decision-making structure which consists of 'fit' individuals (each trained under a different environment scenario), this replaces the conventional best-of-run approach based on a single individual. Committee learning (or ensemble learning) has been adopted widely in statistical-based machine learning methods[64, 43]. Figure 3.3 depicts the key features of the algorithms.

The advantages of robustness and the minimisation of solution risk that accrue from using a team of solutions instead of a single model are particulary suitable for a highly dynamic environment. The advantages that have been previously reported are:

1. Firstly, since the final decision is a combination of a number of problem solvers, one obtains a more consistent estimate of the output. The performance of the system is more robust as the outcome no longer depends on the accuracy of one single model, but on the outcome of several models [77, 151].

2. Secondly, the spread or variance of the different outcomes can be used to derive a measure of confidence, called 'model disagreement indicator'. A small difference in behaviour gives the users greater certainty about the decision [184].

3. Another advantage of a committee is that it enables redundancy. If the committee consists of models that behave differently on different environmental inputs, there will be at least one model available for a particular type of environment [44, 24].

- **Committee decision making**

  In the proposed system, a committee makes decisions by combining the results of several individuals generated by GP runs. The basic idea behind this approach is that GP are population-based and produce a variety of individuals. The diversity of the individuals generated during the evolution provides a rich resource for building committees.

  The decision-making committee is composed of three or more individuals trained independently in a set of diverse market environments. This is to maximise the chance of phenotypic heterogeneity among the committee members. The committee normally comprises an odd number of individuals and the team solution for an environmental case is achieved through the 'simple majority voting' mechanism. Weighted voting is not used in this case because it is difficult to determine which team member contributes the most to the team's overall performance in dynamic environments in order to assign appropriate credits for members.

3.4 Algorithm Implementation

This section describes implementation details of the new algorithms.

3.4.1 Behavioural diversity preservation GP

In this system, each individual has not only a fitness value but also an individual behaviour vector storing its behaviour pattern. How a behaviour pattern is represented in an algorithm is domain specific. But in general an application domain can be classified as non-time dependent (e.g. artificial ant problem, block-stacking problem given by Koza [93]) or time-dependent (e.g. financial time series problem).

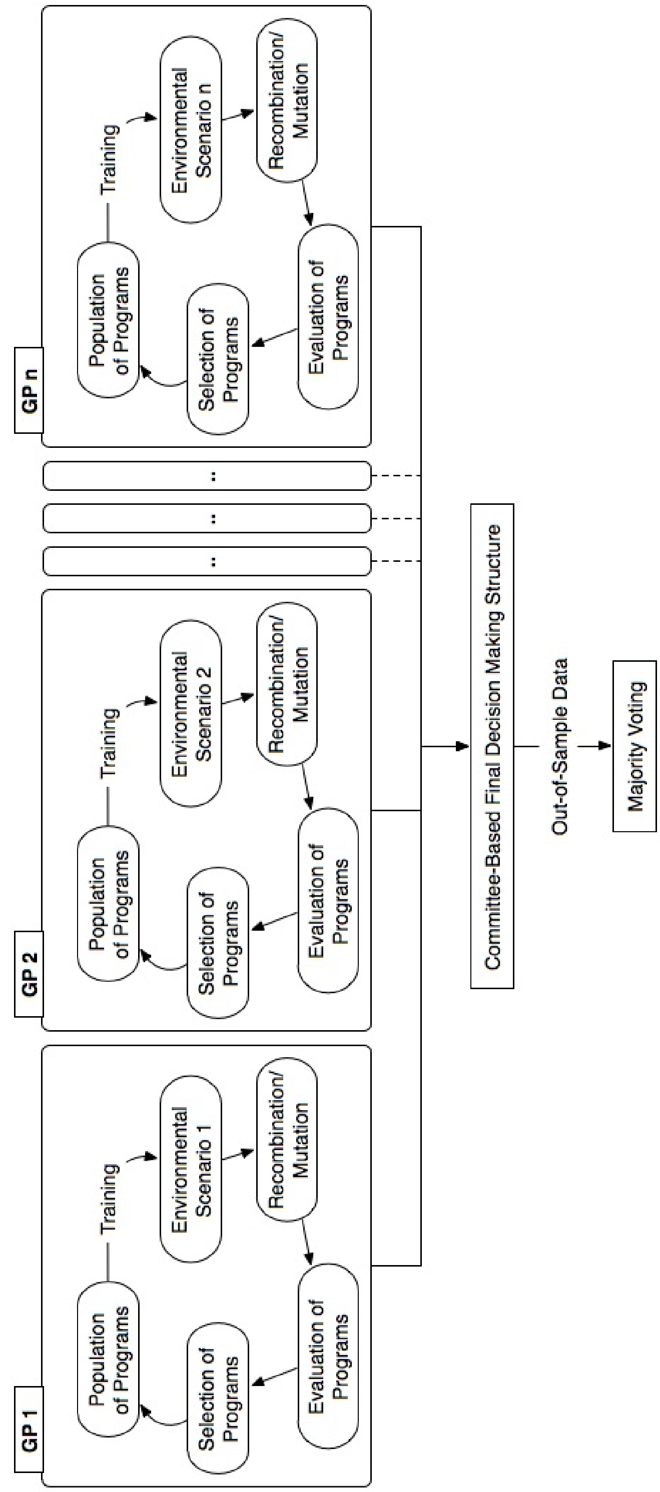


Figure 3.3: Committee Voting GP

- For non-time dependent problems, a large set of training example cases representing a variety of possible problem configurations is usually provided to evaluate fitness of individuals. The behaviour pattern for an individual I, in this context, represents a set of performances recorded for each of the training cases.

  Let C be a vector containing a set of training cases and $C = \{c_1 \ldots, c_n\}$ where c_n is a training case. Let $O(I)$ be the vector of performance of I across C and $O(I) = \{p_{c_1}(I), \ldots, p_{c_n}(I)\}$ where $p_{c_n}(I)$ is the performance of I over a test case c_n . $O(I)$ shall be called the *individual behaviour vector.*

- For time series problems where the training data are measured typically at successive times spaced at uniform time intervals, an individual is trained over a time period. The behaviour pattern for an individual I, in this context, represents a set of historical performance recorded over a time period.

  Let $O(I)$ be the vector of historical behaviour of an individual I during the whole training time period and $p_{t_n}(I)$ be the behaviour of an individual at a particular time t_n. Then, $O(I) = \{p_1(I), ..., p_{t_n}(I)\}$. $O(I)$ shall be called the *individual behaviour vector.*

At the beginning of each evolutionary selection cycle, after the initial population fitness is calculated, individuals are grouped according to their fitness value into a number of segments. Let F_{max}, F_{min} be the maximum and minimum fitness values for a population. Segments $\{(F_{min}, F_{min} + d), (F_{min} + d + 1, F_{min} + 2 * d), ..., (F_{min} + n * d + 1, F_{max})\}$ are collections of fitness intervals of equal length with defined lower and upper bounds. In a fitness segment S, there are n individuals $I_{S_1} ... I_{S_n}$ each with a behaviour vector.

Behavioural diversity in each of the fitness segments can be determined by measuring the correlation of the behaviour vectors either between an individual and every other individuals in its segment or between an individual and its segment. The major advantage of the second method is the computational cost ($O(n)$ rather than $O(n^2)$). The segment behaviour vector can be defined as a vector of average mean performance output of all individuals in a fitness segment S over a time period (for time series problems) or over a number of cases (for non-time dependent problems) t_i.

If the difference of behaviour pattern between an individual and other individuals or its segment is large, then it means that this individual exhibits unique or largely different behaviour from other individuals. The difference is measured based on the Spearman correlation test between two individuals I_x and I_x. The Spearman correlation measure simply ranks the two variables, and makes no assumption about the distribution of the values. The Spearman correlation coefficient $\rho(O(I_x), O(I_y))$ is computed as follows:

$$1 - \frac{6 \sum_{i=1}^{N} d_i^2}{N^3 - N} \tag{3.1}$$

where N is the numbers of pairs , and d_i is the distance between (i) the rank of performance $p_i(I_x)$ (compared with all other $p_j(I_x)$, $j \neq i$) in the individual behaviour vector $O(I_x)$ and (ii) the rank of $p_i(I_y)$ (compared with all the other $p_i(I_y)$, $j \neq i$) in another behaviour vector $O(I_y)$. The degree of correlation returned by this measure varies from -1.0 representing negative correlation, through 0.0 indicating no correlation, to 1.0 representing positive correlation. As an example, consider three individuals (i.e. trading strategies) X, Y and Z in a segment with behaviour vectors as follows:

$O(I_X) = (-20\%, +10\%, -20\%, +40\%)$

$O(I_Y) = (+3\%, +2\%, +2.5\%, +2.5\%)$

$O(I_Z) = (-3\%, -2\%, +6\%, +0.5\%)$

Then the segment behaviour vector S would be

$S = (-6.67\%, +3.33\%, -3.83\%, +14.33\%)$

And the behavioural differences between the segment vector S and each of $O(I_X)$, $O(I_Y)$, and $O(I_Z)$ is given by the Spearman correlation coefficient as:

$\rho(S, O(I_X)) = 0.9$

$\rho(S, O(I_Y)) = -0.5$

$\rho(S, O(I_Z)) = 0.4$

From the above rank correlation coefficients, it may be deduced that individual Y exhibits unique or largely different behaviour from other individuals.

If a segment contains a set of individuals that have correlated or similar behaviour throughout the run, the behaviour diversity of the segment would be compromised without deleting individuals from this set. Conversely, if individuals are kept, whose reactions to the environment are very different from other individuals, the diversity level of the segment can be maintained. Therefore, there is firstly a check for the correlation coefficient value ρ of each individual in each segment and then individuals with high ρ (higher than a predefined correlation threshold, which has been be set at 0.67) are deleted. Additionally, in the case of domination of one or two particular fitness segment(s)- those where the number of individuals contained in the segment are higher than the average (defined as P/s where P is the total population size and s the number of segments)- not only shall those individuals with a high correlation coefficient be deleted, but also surplus individuals in order of decreasing correlation. In this way, the algorithm encourages the creation of individuals at all fitness levels throughout the evolutionary cycle and also preserves diversity at the global population level. After crossover and mutation, randomly generated individuals are inserted into the population in order to keep the population size constant.

3.4.2 Multiple-scenario GP

The investigation is concerned with not only the performance or fitness of the GP-evolved solutions, but also the performance consistency of the GP-evolved solutions across a range of environment dynamics. For example, in a financial scenario where market prices are rising ('bull market'), a scenario where market prices are falling ('bear market') and a scenario where market prices are fluctuating with large amplitude ('volatile market').

The training data is therefore considered to be a set of fitness cases — a vector of environments — representing a possible range of different environments and then adjust the fitness with its perceived volatility to obtain an whole picture of an individual's performance .

There are three ways in which the GP population can be exposed to these scenarios:

1. 'Standard GP' (SGP): use the entire vector S, treated as a single unit, throughout all generations;

2. 'Multiple-scenario Evaluation in the Last Generation' (MELG): use a variant of the three-dataset methodology [129, 58], where the entire vector S (treated as a single unit) is used for $n-1$ generations, and in the final generation individuals are tested against a subset of environments $\{s_i\}$ drawn from S. The 'best-of-run' individual used in the validation on set V is that which has, in the final generation, the highest 'Consistency-Adjusted Fitness' (CAF)(see below);

3. 'Multiple-scenario Evolution' (MEVO): in each generation, use a subset of environments $\{s_i\}$ drawn from S, and ascribe to each individual a CAF (see below). Evolution proceeds as normal on the basis of this adjusted fitness measure.

Consistency-adjusted fitness

S has previously been introduced as the training environments vector and s_n as the n^{th} possible scenario; hence $S = \{s_1 \ldots, s_n\}$. Now let $M = \{m_1 \ldots m_p\}; m_j \in S$ be a subset of S that is used for fitness evaluation.

Let I_i be an individual in the population, and $f_{I_i}^{m_j}$ be the fitness of individual I_i when evaluated on scenario m_j. Then F_{I_i} is the 'fitness vector' of I_i when evaluated on a subset of scenarios, given by $F_{I_i} = \{f_{I_i}^{m_1}, \ldots, f_{I_i}^{m_p}\}$.

Standard deviation is used to calculate the consistency of the fitness (performance) of the individual across this range of scenarios:

$$\sigma_{I_i} = \sqrt{\frac{1}{p}\sum_{j=1}^{p}(f_{I_i}^{m_j} - \overline{F_{I_i}})^2} \tag{3.2}$$

where: $\overline{F_{I_i}}$ is the mean of F_{I_i}, given by $\frac{1}{p}\sum_{j=1}^{p} f_{I_i}^{m_j}$. The 'Consistency-Adjusted Fitness' (CaF) of an individual is now defined as the mean fitness divided by volatility:

$$CaF_{I_i} = \frac{\overline{F_{I_i}}}{\sigma_{I_i}} \tag{3.3}$$

MELG

For MELG a variation of the three-dataset methodology is used, as outlined in [129, 58]. In this version of the methodology, the *training set* is used to evaluate the fitnesses of individuals in $n-1$ generations; elitism ensures that the best-of-generation individuals survive to the last generation, and the individuals in the last (n^{th}) generation are tested against a different *in-sample volatility set*; a best-of-run individual is selected and its quality is assessed using yet another, different, *out-of-sample validation set*.

Note that a possible drawback of this methodology is that, where data samples are limited, either the training set or the out-of-sample validation set must be smaller than it would be in a two-dataset methodology. However, in this proposed variation of the methodology, the *in-sample volatility set* is a vector of subsets of the initial *training set*. The fitness vector was also chosen so as to be set at $F_{I_i} = \{f_{I_i}^{m_0}, f_{I_i}^{m_1}, \ldots, f_{I_i}^{m_p}\}$, where $f_{I_i}^{m_0}$ is the fitness of the individual previously calculated in the $n-1^{th}$ generation — thus, the fitness vector contains information about fitness for the whole training set treated as a single unit, as well as about fitness relating to each of the scenarios.

The selected methodology permits a direct comparison with SGP, since it is known that both SGP and MELG have been given identical training data — what is different is the way in which that data is presented to the population.

MEVO

The MEVO algorithm differs from MELG in that it uses a two-dataset method: the *in-sample volatility set* used in MELG is used not only in the final generation, but in every generation. The second dataset is therefore the out-of-sample validation set V.

Thus, evolutionary selection is based on the consistency-adjusted fitness CaF_{I_i}, which is calculated from $F_{I_i} = \{f_{I_n}^{m_1}, \ldots, f_{I_n}^{m_n}\}$ (see above). F_{I_i} can be thought of as an 'intermediate' fitness vector for each individual, and VaF_{I_i} is the 'real' fitness of an individual. Note that MEVO is *not* exposed to the entire training set ($f_{I_n}^{m_0}$); it is only exposed to the extreme scenarios. One might think that MEVO would thus be put at a disadvantage, because it does not have access to as much information as MELG or SGP, but in early trials it was discovered that using the entire training set as another scenario led to poor results.

3.4.3 Committee voting GP

A committee C is composed of a number of members $(I_1 \ldots, I_n)$ which are evolved by entirely separate GP evolutions using different training data to obtain widely-differing behaviour in the committee; hence $C = \{I_1 \ldots, I_n\}$. In a new environment, C generates an output vector o with $o = \{o_{I_1} \ldots, o_{I_n}\}$ where o_{I_n} represents the output generated by each member in the C. Then o needs to be combined to obtain a single output as a final solution O for the algorithm. This can be expressed as $O = f(o)$ where f is a combination function.

This algorithm uses a simple f function because its essential novelty is using a committee not a novel combination approach which is of great focus for ensemble methods research for machine learning [43] and is not in the scope of this thesis.

Depending upon the types of application in real-world, o_{I_n} may be required to generate either (a) numerical value(s)(e.g. a temperature prediction; a stock price prediction or a set of parameters values for architecture) or (a) choice(s) among many (e.g. buy or sell signals for trading systems or fraud detection or a set of robotic instructions).

In the first case, if $o_{I_n} \in \Re$, then f is a simple average function:

$$O = \frac{1}{n}\sum_{i=1}^{n} o_{I_n} \tag{3.4}$$

In the second case, the f is a simple majority voting function; more precisely, the final solution is accepted if it is voted for by more than the half of the members of C.

Because this thesis applies the algorithm in a financial trading context involving either buying or selling stocks, the simple majority voting function is used in the implementation and more details are explained in Section 5.3. It does not exclude the usage of a simple average function in other applications.

3.5 Summary

This chapter has described the key features and implementation details of the three new algorithms designed to enhance robustness of solutions evolved by GP.

A new categorisation method of dynamic environments developed by this work was presented at the beginning of the chapter. The new categorisation aimed to underline key characteristics which could be associated with different types of environments and to facilitate algorithm design and evaluation, which needed to take into consideration these characteristics. In accordance with four characteristics of changes, such as frequency, predictability, morphology and visibility, dynamic environments were divided into categories Type 1, Type 2, Type 3 and Type 4. Type 2 and Type 3 environments are characterised by frequent, unpredictable and large changes and are extremely common in real world. The novel algorithms presented in this work were specifically designed for Type 2 and Type 3 environments.

The first algorithm, ‘behavioural diversity preservation’, is a novel diversity preservation technique for use in a Type 2 dynamic environment. The algorithm evolves more robust solutions by preserving population phenotypic diversity through the reduction of their behavioural intercorrelation and the promotion of individuals with unique behaviour.

The second algorithm, ‘multiple-scenario training’, is a novel population training and evaluation technique for use in Type 3 dynamic environment. The algorithm evolves more robust solutions by training a population across a set of preconstructed environment scenarios simultaneously and by using a ‘consistency-adjusted’ fitness measure to favour individuals performing well across the entire range of environment scenarios.

The third algorithm, ‘committee voting’, is a novel ’final solution’ selection technique for use in Type 3 dynamic environment. The algorithm enhances robustness by breaking away from ‘best-of-run’ tradition, creating a solution based on a majority-voting committee structure consisting of individuals evolved in a range of diverse environmental dynamics.

The next chapter describes the case study for robustness evaluation of these new algorithms.

Chapter 4

Case Study - a long-short hedge fund in the Malaysia equity market

In the previous chapter, a detailed description of new algorithms was provided. In order to evaluate the robustness performance of these techniques, an appropriate, challenging and interesting real-world validation environment is needed in which to apply them. In this chapter, a real-world case study environment is introduced - a Malaysian equity market, a description of a simulated investment strategy system powered by GP is given and the experimental method and detailed settings for performance evaluation are clarified.

4.1 Malaysian Equity Market

Emerging stock markets have for many years attracted market participation due to their return potential and higher return predictability, than developed markets. Malaysia is one of the biggest emerging economies in east Asia and its stock market, the Kuala Lumpur Stock Exchange (KLSE), was at various times the most active exchange in the world, with trading volume exceeding even that of the New York Stock Exchange (NYSE). The Kuala Lumpur Composite Index (KLCI) is the leading stock market indicator.

As well as possessing the many environmental characteristics, discussed in Section 2.4, the Malaysian stock market has high volatility with continuous irregular changes and a high frequency of large morphological changes. This is in line with the overall patterns exhibited by emerging markets with Type 2 and often Type 3 environments dominating at different time periods. These characteristics are easy to detect by inspection of Figure 4.1 showing KLCI index movements during the 1996 to 2005 period.

The environment was particularly volatile and uncertain from mid-1997 to mid-1999, with sharp falls and rises. This reflected the direct impact of the east Asian financial crisis, as the economy and the confidence of the investor were badly shaken. During this period, it was a Type 3 environment. It can also be observed that there were many large changes which generated the notable environmental morphological shifts. These changes were influenced by important domestic and international political and economic events (marked as the circles in the graph). The domestic events included the capital control measures implemented by the government in

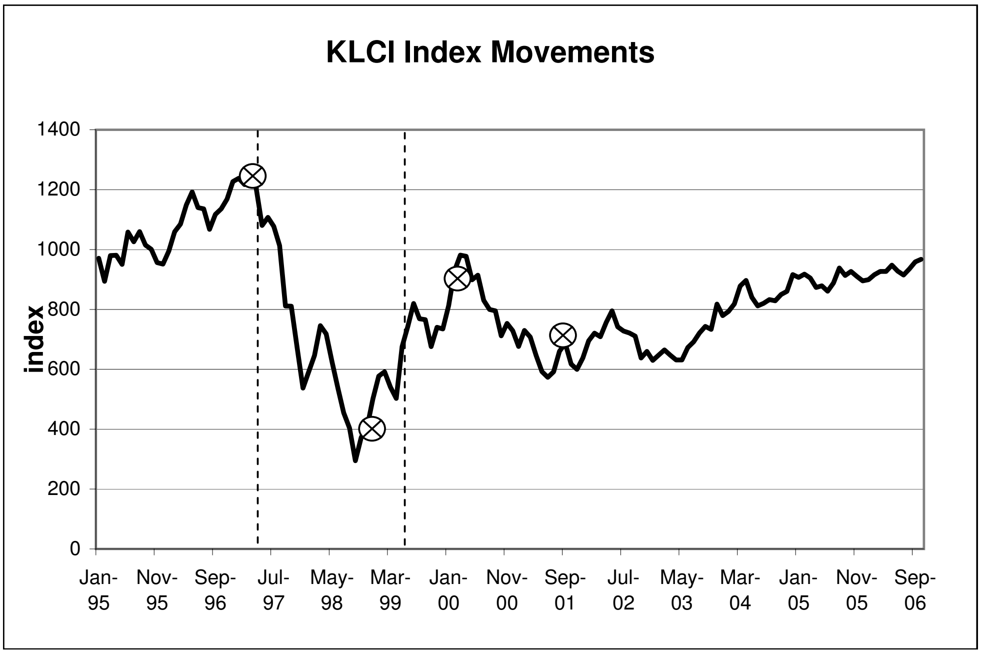


Figure 4.1: Malaysia Stock Market Environment

September 1998 that set off the market recovery, and the cooling down of the property bubble at the beginning of 2000 that marked a period of market down-turn. International events included the 1997-1998 Asian financial crisis, which triggered a long period of high volatility, and the 11th September 2001 tragedy that suddenly but briefly interrupted the market recovery and stabilisation process.

4.2 Hedge Fund Simulation and GP system

In this thesis, a long-short hedge fund is simulated and applied to the Malaysian stock market, with the objective of generating a return *entirely* from the stock selection model provided by a GP system.

The hedge fund simulation adopted by this work consists of three steps:

1. Identification of stock price related factors for inclusion in the stock selection model.
2. Development of portfolio investment simulator.
3. Development of GP stock selection system.

4.2.1 Factor Selection

The choice of factors is influenced by the work of Achour et al. [5]. Their work is probably the most comprehensive analysis so far concerning firm attributes for investment in emerging markets (including Malaysia). Factors are classified into three groups: historical accounting characteristics (fundamental factors); past returns (technical factors) and size (size factors). Each of the factors is detailed in Table 4.1

Factors	Definitions	Interpretations
Market capitalisation	number of shares * closing monthly market price	•Diagnostic screen to investigate performance differential between large and small capitalised stocks. •Capitalisation (size) is widely regarded as a proxy for trading liquidity. •Risk, as defined by volatility of historical returns, tends to increase as size decreases.
Change in return on equity	return on equity (current year) - return on equity (previous year)	•The objective is to identify companies that investors believe are of higher 'quality'.
Debt to common equity ratio	(total debt / common equity) *100	•Diagnostic screen constructed to give insight into performance differential between leveraged and unleveraged stocks. •Debt/equity ratios can be used as a proxy for 'quality' and perceived risk and screens on 'good' and 'bad' companies.
Dividend yield	(sum of last 12 months of cash dividends / closing monthly market price)*100	•Higher yielding stocks may exhibit superior performance over time.
6, 12, 36 months historical earnings growth / momentum (G/M)	(last 12 months' trailing earnings per share - previous last 12 ms trailing earning per share) / (absolute previous last 12 ms trailing earning per share)*100	•Earning momentum indicator frequently used as the best growth proxy due to information deficiencies in emerging markets. •Useful indicator to identify those stocks with rising expectations among investors prior to their establishing a track record.
One year historical earnings growth rate	the rate of change in the reported last twelve-month earnings per share over the three year time interval (terminating on the date of the last interim period for which earnings were announced.)	•A traditional growth proxy highlighting a stock's historical track record and stability. •Stocks which pass factor criteria have a visible track record, a perceived rarity in the volatile emerging markets, and should therefore trade at high premiums even though it is generally accepted that naive extrapolations in these volatile markets are futile.
Earnings yield	(last 12 ms' trailing earnings per share / closing market price)*100	•Traditional 'value' / 'growth' proxy used by investors. •Value stocks generally are riskier as they are usually firms under distress, have high financial leverages and face substantial uncertainty in future earnings.

Book to price ratio	(historical book value per share / closing monthly market price)*100	•Traditional 'value' / 'growth' proxy. •Conventional wisdom suggests that the book-to-price ratio is one of the most straightforward and effective investment factors in the emerging markets.
Cash earnings to price yield	(cash earnings per share / closing market price)*100	•Traditional 'value' proxy. •It provides some information regarding a company's ability to leverage itself, to pay dividends and enjoy financial flexibility.
1-month, 1-yr price momentum	one month/year price change	•A firm's past returns might help to predict future returns.
Revenue growth	[(current year's net sales or revenues / previous year's net sales or revenues)-1]*100	•Revenue growth is often used as a proxy for 'quality' and real short-term 'growth'.
Rate of reinvestment	(last 12 months' trailing earnings per share - last 12 ms dividend per share)/ (last year's book value per share)*100	•The ratio used to discriminate 'growth' companies that provide higher rates of returns on invested capital but reinvest earnings to generate internal growth rather than returning capital to shareholders. •If the firm has good prospects, a high reinvestment rate would be expected.
Return on equity	(last 12 months' trailing earnings per share / last year's book value per share)*100	•Traditional 'quality' and risk proxy to investigate the performance differential between perceived 'good' and 'bad' stocks over time.
26/12 day Moving Average Convergence-Divergence (MACD)	Difference between 26-day and 12-day exponential moving averages	•Traditional technical price trend indicator. •Positive and rising MACD indicates a bullish period for the price. •Negative and declining MACD indicates a bearish period for the price.
30-day simple moving average	30-day price moving average	•Traditional technical price trend indicator. •It smoothes out a data series, indicating the direction of a price trend.

Table 4.1: Factors Definitions and Interpretation

4.2.2 Portfolio Investment Simulator

The portfolio simulator simulates the portfolio investment strategy adopted by the hedge fund. A key characteristic of the strategy is market neutrality, which is implemented as follows:

- The portfolio is dollar-neutral, which means it maintains equal dollar amounts of long and short investment to achieve a balanced long/short exposure. For convenience, the global portfolio can be understood to consist of a long sub-portfolio and a short sub-portfolio, equally weighted.

- The portfolio is sector-neutral, which means it maintains investment balance across the four main market sectors (financial, retail, property and manufacturing) and within each sector; the portfolio remains evenly divided between stocks. Being sector-neutral, the portfolio attempts to offset the risk of sector/industry-wide volatility.

- Before the global portfolio is constructed, the stocks are grouped into market sectors and within each sector all stocks are ranked according to expected return, which indicates the attractiveness of the stock. Each sector of stocks is then divided into top stocks, middle stocks and bottom stocks. The top stocks, which have the highest rankings, are assigned to the 'long' portfolio. The bottom stocks have the lowest rankings and are assigned to the 'short' portfolio.

The portfolio investment takes a monthly trading approach (as opposed to high-frequency trading strategy) to minimise transaction costs and taxes. Stocks in the long-short portfolio are subject to a monthly investment review. At the end of each month, all of the positions held in the portfolio are closed and the profit or loss of the portfolio during the month is calculated. Although closing all the positions at the end of each month is not a real hedge fund trading practice, it facilitates accounting during programming. This leads to a slightly conservative estimate of GP performance as the costs of closing the positions represent a very small percentage indeed. At the end of each year, the Sharpe ratio is calculated.

The profits of the overall fund portfolio are derived from the performance spread between the stocks held long and the stocks sold short [1]. These profits are in addition to the interest received on proceeds of the short sales as the stock borrowed does not need to be returned until a later date.

4.2.3 GP System

Selection of the stocks which form the long-short portfolio for the investment simulator is entirely guided by the GP system. This is implemented by coupling the investment simulator and the GP, as illustrated in Figure 4.2. The coupling between the two systems is the fitness function: the investment simulator is called each time the GP system needs to determine the fitness of an individual. The solution model evolved by the GP is a non-linear combination (equation) of stock return factors, described in the previous section.

[1] CFDs (Contracts for Differences) are used instead of conventional shares to trade on stocks.

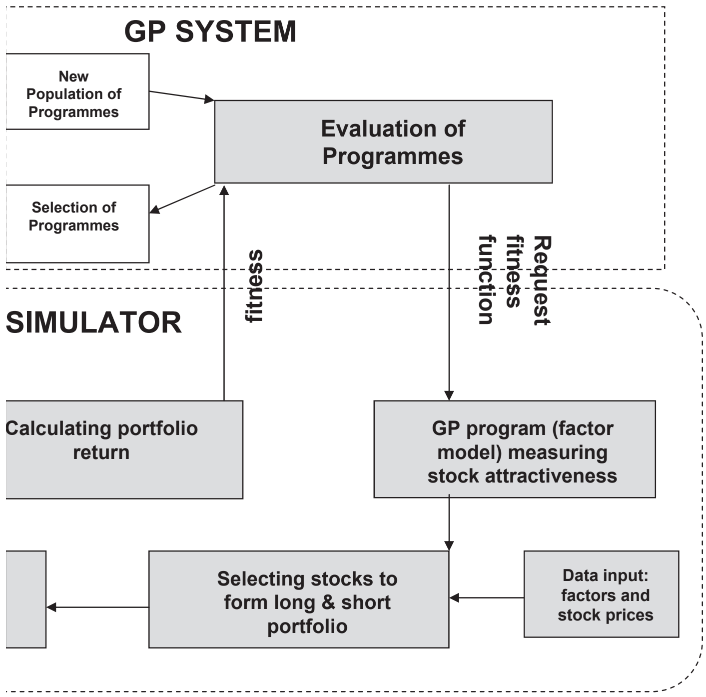


Figure 4.2: Validation System Design: GP + Investment Simulator

The GP process comprises the following steps:

1. Randomly generate an initial population of stock return models. Each GP individual is a nonlinear mathematical equation composed of the factors described in the previous section, a range of arithmetic functions and real value constants. The GP-evolved equation generates a quantitative measure for the investment simulator to rank and select the stocks.
2. Calculate the fitness measure of each individual.
   (a) Iteratively perform the following sub-steps for each simulated sub-period time (e.g. month by month) until the end of the simulated investment period:
      i. For each stock in the investment universe, the Investment Simulator retrieves appropriate variables of the nonlinear function, i.e. monthly factors and price data from a stock information database (see data section), and then calculates the equation. The result of the equation is a quantitative measure of the attractiveness of a stock.
      ii. The investment simulator then constructs the portfolio, sells and buys the stocks selected and also computes the overall profit and loss (P & L) of the investment during the trading sub-period.
   (b) Return the computed Sharpe Ratio across all sub-periods as the fitness measure.
3. Use the fitness measures to select models from the population to create new individuals for the population by applying reproduction, crossover and mutation.

4. Repeat until the termination criterion is satisfied.

4.3 Data

Data for the universe of stocks as well as benchmark returns are sourced entirely from the Reuters 3000Xtra database system. The relevant investment universe focuses on the companies listed on the Kuala Lumpur Composite Index, which is the main index for the Malaysian stock market and is comprised of 100 companies in total. The universe actually contains 36 company stocks, and that is solely because of limited data availability from the Reuters system.

For every stock in the universe, data consists of the monthly closing stock prices, expressed in US dollars as given by Reuters, and monthly data for 19 factors. Monthly rather than daily data were chosen to avoid the potential bias associated with microstructural issues, non-trading and the problem of thin trading [2], all of which are often associated with most emerging markets. Data span from the period of 31st July 1997 to 31st December 2004.

4.4 Experiment Method

This section gives a description of the experimental conditions used to evaluate the proposed techniques. These conditions have been applied to all experiments conducted in this thesis and are described below:

- The simulation of a long-short equity fund in a Malaysian equity market is used as a dynamic environment case study for all experiments.
- Robustness of the solutions/individuals evolved by GP algorithms is evaluated in accordance with the robustness definition described in Section 3.2. Therefore, there are 3 measures.

  1. fitness (F) which is calculated depending on the different methods designed for each algorithm as described in Section 3.4 and the next chapter.
  2. standard deviation of fitness (σ) measured at each time interval unit.
  3. percentage of decrease in fitness between training and validation (c)

  Robustness of individuals evolved by the new algorithms are compared against those evolved by a canonical GP algorithm. In order to put the comparison in a real world context, the results are also benchmarked against the traditional investment strategies often used by portfolio managers (e.g. buy and hold strategy (index) or technical trading strategies).

  An individual (I_1) is more robustness than another individual (I_2), if:

  1. $F(I_1) > F(I_2)$ when $\sigma(I_1) \approx \sigma(I_2)$, or,

[2] "A condition in which there is little trading activity in a market because of a lack of buy or sell orders to drive up the volume. Thin trading usually occurs around holidays and sometimes in the doldrums of August. Thin trading conditions make it difficult for large buyers or sellers to execute orders because their trading activity may move prices." [49]

Population size (N)	1000
Method of generation	Ramped half and half
Function set	{+, -, *, /, Exp}
Terminal set	18 firm-specific factors
Selection scheme	Fitness proportionate selection
Criterion of fitness	Monthly Sharpe ratio
Number of trees generated by elitism	10 (1%)
Number of trees generated by crossover	950 (95%)
Number of trees generated by mutation	40 (4%))
Termination criterion	100-generation evolution
Maximum depth of initial generation	6

Table 4.2: GP Parameter Settings

2. $\sigma(I_1) < \sigma(I_2)$ when $F(I_1) \approx F(I_2)$, or,
3. $c(I_1) < c(I_2)$

- The following investment simulator parameters are used across all experiments.
  - The stock universe is grouped into 4 main market sectors.
  - At all times, 12 long positions and 12 short positions were maintained, thus there are in total 24 stocks in the portfolio and each counts for approximately 4% of the total portfolio value.
  - According to the ranking, the top 3 stocks in each sector became the top fractile (for long positions) and the bottom 3 became the bottom fractile (for short positions).
  - In accordance with common industry practice, there is an assumption of 20% notional trading requirement (margin), 0.25% trading commission (trading cost) and 5% financing rate.
  - The initial investment capital was set as US$100M to keep the size of the fund relatively small and flexible enough to implement the investment strategy.
- The comparison of the new techniques is conducted across 25 runs, which is the chosen sample size. A larger sample size (more runs) would provide slightly more precision in the comparisons, but the trade-off would be the extensive use of computational resources. A small sample size, on the other hand, would provide less precision. Hence the choice of 25 was made because it provides a reasonable sample size from which a fair comparison can be made, and also saves on computational resources.
- For all 25 independent runs, the same set of seeds (for the random number generator) is used for each of the techniques being tested, under the assumption that the initial population of each GP run will have a significant effect on the population.

- To make this a strict comparison, all GP operators and parameters were kept constant. To be consistent, all experiments and analysis in this work were conducted using these conditions, which represent one variation of the many different forms of GP. The parameters listed in Table 4.2 were used.

- ECJ[3] GP package is used to implement all the GP systems including standard GP and three new GP algorithms. The parameters chosen for the experiments are default settings for ECJ environment that are widely adopted by the research community.

4.5 Summary

This chapter presented the Malaysian stock market as the chosen case-study environment for evaluating the proposed techniques. It is a challenging environment for GP as it has relatively high volatility and frequently undergoes large changes.

This chapter also described the design of the long-short hedge fund simulation system. The system comprises an investment simulator coupled with a GP system. The GP system evolves non-linear factor equations that guide the ranking process performed by the investment simulator during the stock selection process to form a long-short portfolio.

Finally, this chapter also clarified a set of experimental conditions used for evaluations of the algorithms. These conditions include robustness measures in the context of the financial application, fitness measure, investment simulator parameter settings, GP parameter settings and number of GP runs.

The next chapter evaluates the three proposed algorithms (namely a behavioural diversity preservation GP, a multiple-scenario GP and a committee voting GP) in terms of robustness performance, and compares them with the canonical GP approach.

[3] A Java-based evolutionary computation programming environment developed at George Mason University's ECLab Evolutionary Computation Laboratory.

Chapter 5

Evaluation Results

In the previous chapter there is a description of our dynamic environment case study for robustness evaluation of the proposed technique. In this chapter an evaluation is conducted of the 'multiple-scenario' approach, the 'committee voting' approach and the 'behavioural diversity preservation' approach. This work has been published in [175, 177, 174, 179]

In order to verify the hypothesis of this thesis, the three new GP approaches are empirically evaluated by comparing their performance, robustness in particular, with that of the canonical GP on a real-world application problem: stock selection in long-short hedge funds in an emerging market. The results of experiments should determine:

- the efficacy of the new system — that is to say, to what extent it improves performance (fitness) in general, compared with a standard GP system;
- the robustness behaviour of an individual chromosome obtained by such a system, compared with that obtained from a standard GP system.

This chapter, for each algorithm, firstly specifies the experiment settings (i.e. experiment method, key research questions and data) and then offer detailed discussion and analysis on the experiment results.

5.1 Behavioural Diversity Preservation GP

5.1.1 Experiment Settings

The motivation for this experiment is to validate whether the behavioural diversity preservation GP can be used in a continuous-learning context [42] which is particularly suitable in a Type 2 environment. Continuous learning in this context refers to retraining individuals whenever the market environment changes and a new batch of training data is available. The previously trained population is retained so that it can be retrained on this new data. Old data is not used for retraining.

Following a shift in economic context it will be necessary to continue to use the previously trained 'best' individual while new data is being accumulated for retraining — the interest is in the behaviour of this previously-trained 'best' solution, and how well it performs in the context of a new economic environment. When sufficient new data has been accumulated for retraining, it is required that the system should retrain effectively on the new data — the expectation is that shifts in context will not normally be excessive in a Type 2 environment and therefore retraining should start from the previously trained population (rather than from a random population) since the system is then less likely to converge on a local optimum, especially when there are few data points available for the new environment [133]. Initial training will use an 'in-sample' data set; subsequent retraining will be based on 'out-of-sample' data in the context of the original training.

Experiments were performed to answer the following three research questions:

1. does the new technique really improve population diversity when compared with a standard GP system, and how does this affect fitness?
2. when retraining on data that comes from a different economic context, how quickly, and how well, does the diverse population adapt to a new environment?
3. are trained individuals from the new system more robust when exposed to a new economic environment?

Three new experiments were run, presented below. In each case, a standard GP system (SGP) was compared with the new GP system (NGP). The training data and validation data was in all cases identical for both SGP and NGP.

The fitness f for an individual is given by Equation 5.1.

$$f = \frac{1}{1 + |1.5 - S|} \quad \text{where} \quad S = \frac{1}{n}\sum_{i=1}^{n} \frac{x_i - RFR_i}{\sigma_i} \tag{5.1}$$

In Equation 5.1, S is the average Sharpe Ratio over the training period (comprising n sub-periods), x_i is the monthly ROI over the sub-period i, σ_i is the standard deviation of monthly ROIs over the sub-period i, and RFR_i is the average monthly Risk Free Rate for sub-period i. RFR_i was set to 0.003 for all i (equivalent to 3.6% per annum). Fund managers often set a target Sharpe Ratio, as shall these experiments — the target set is 1.5 [1], and the absolute difference between the measured Sharpe Ratio and the target is then normalised to provide a fitness value that varies between 0 and 1.

In-sample population dynamics (Experiment 1)

The aim of Experiment 1 was to see whether NGP improved diversity, and what effect it had on fitness. Both SGP and NGP were trained separately using 41 months of data for 24 Malaysian stocks (from 31/7/97 to 31/12/2000). The following measurements were made for each generation:

1. the fitness of the best chromosome, the fitness of the worst chromosome, the average fitness across all chromosomes, and the standard deviation of all fitnesses;

[1] Achieving Sharpe ratio of 1.5 is considered as an industry excellence by the fund managers for a long-short hedge fund. For this reason the GP fitness function adopted this industry benchmark as the target.

2. the distribution of chromosomes in the segment vector in terms of (i) fitness and (ii) phenotypic behaviour.

The average value from 25 runs of each measurement was plotted for each generation.

Retraining population dynamics (Experiment 2)

The aim of Experiment 2 was to see how well the population as a whole adapted to a new environment; in each case it started with a population that had previously been trained on the 41 months of training data (Experiment 1 above) and then retrained SGP and NGP separately on new training data that reflected a different economic climate. Three sets of new training data were used and the population behaviour for each was investigated:

Period a: 31/12/2000 — 31/4/2002 (16 months).

Period b: 31/4/2002 — 31/1/2003 (9 months)

Period c: 31/1/2003 — 31/12/2003 (11 months)

As with Experiment 1, this experiment was repeated 25 times and for each generation the best, average and worst fitnesses, and standard deviation, were measured.

Out-of-sample individual behaviour (Experiment 3)

The aim of Experiment 3 was to investigate whether improved population dynamics has resulted in improved robustness of behaviour of *individuals* in a previously unseen context, using the three robustness measures in both a training-validation context and a continuous-learning context. The following three sets of data were used for this experiment. Period a: 31/07/1997 to 31/12/2000; Period b: 01/01/2001 to 31/06/2002; Period c: 01/07/2002 to 31/12/2003.

1. **Training-validation**. Using the best individual trained using Period a + Period b, a validation test was run on Period c. First, the best trained individual from the SGP system was used, and then the best trained individual from the NGP system.

2. **Continuous-learning.** For both SGP and NGP separately, starting with the previously trained population from Period a, then the population was retrained using Period b. The best individual was obtained and a validation test was run on Period c.

In all cases (for both SGP and NGP) the monthly ROI and monthly ROI standard deviation were measured and recorded, the Sharpe Ratio for the entire test period calculated, and the experiments repeated 25 times.

5.1.2 Discussion of Results

In-sample population dynamics

The first research question posed was *"Does the new technique really improve population diversity when compared with a standard GP system, and how does this affect fitness?"*. Fig. 5.1 illustrates

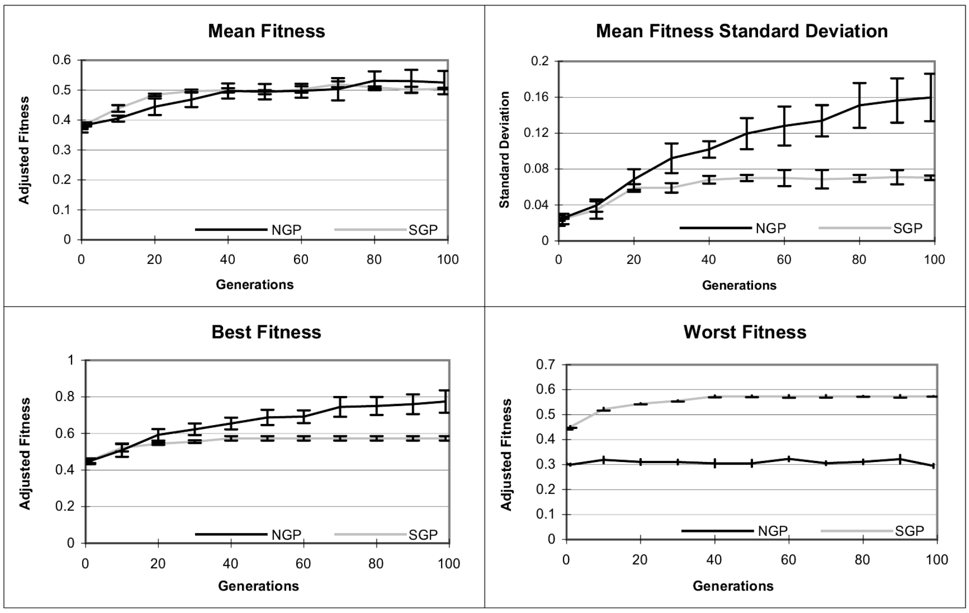


Figure 5.1: In-sample training (population dynamics). The average value from 25 runs of each measurement was plotted for each generation ; the best and worst runs are indicated by error bars.

how the fitness of individuals in the population evolved over 100 generations while being trained on the base data of 41 months. Whilst the mean fitness difference between SGP and NGP remains negligible as the population evolves, the difference in standard deviation of fitnesses clearly increases as the population evolves, with NGP developing a substantially greater diversity of fitness. It is interesting to note a corollary of these two statements, which is that the fitness of the *best* individual is much greater for the NGP system — this in itself is a compelling reason to use diversity-preserving techniques, regardless of other possible benefits in terms of adaptability.

Fig. 5.2 gives the distribution of individuals across the fitness segments ($\kappa = 5$) at three points during evolution (after 0, 50 and 100 generations), indicating a slight bi-modal characteristic for both SGP and NGP. Of more importance from the point of view of 'phenotypic behavioural' fitness, Fig. 5.3 investigates the correlation coefficient of individuals in the highest fitness segment (the segment from which the final solution(s) will be drawn). Fig. 5.3 illustrates how phenotypic behavioural diversity increases as the population evolves — to help visualise this behaviour, on the left is a graph of correlations from a single (typical) run of NGP showing the mean, best and worst correlations of individual behaviour vectors together with the segment behaviour vector. To quantify this behaviour, on the right is a graph comparing the mean correlations with error bars for 25 runs for NGP and SGP.[2] The results indicate that NGP provides a substantial improvement over SGP in diversity of phenotypic behaviour of the highest-fitness segment in the later generations. It takes several generations for the diversity-

[2]For SGP, population snapshots were saved for generations 0, 20, 50, 80 and 100, and those individuals identified which would have been in the highest fitness segment if NGP were being run; then each such individual was used in the investment simulator and its individual behaviour vector obtained; finally, a segment behaviour vector was calculated and the correlations measured (repeated for each "snapshot").

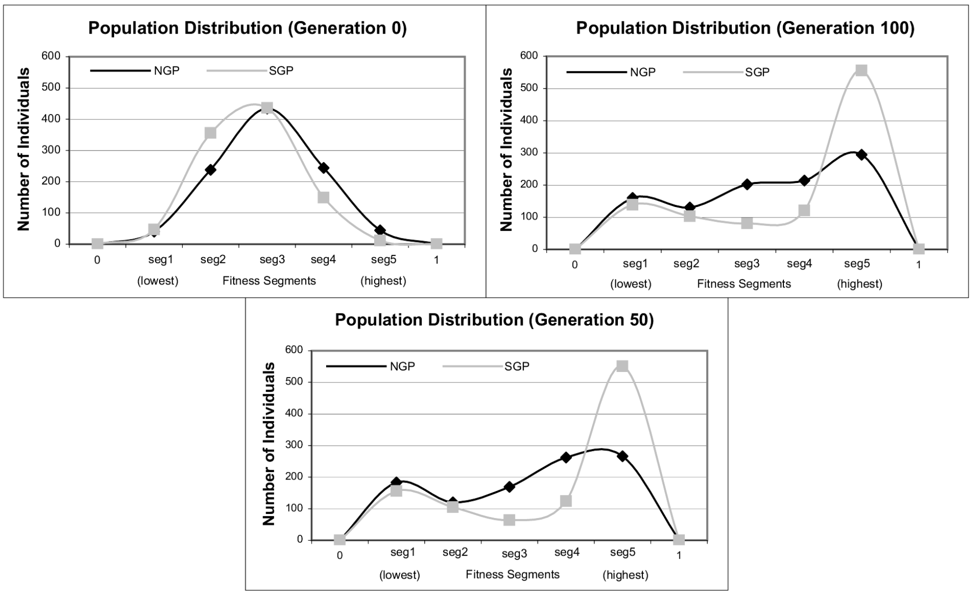
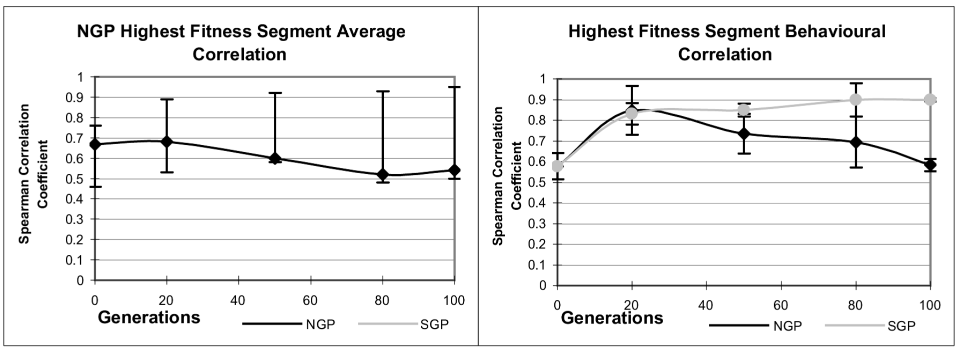


Figure 5.2: Phenotypic diversity for SGP and NGP. The average value from 25 runs of each measurement was plotted for each generation ; the best and worst runs are indicated by error bars.



An example of a particular run of NGP **NGP vs SGP (multiple runs)**

Figure 5.3: Evolution of behavioural phenotypic diversity for SGP and NGP. The average value from 25 runs of each measurement was plotted for each generation ; the best and worst runs are indicated by error bars.

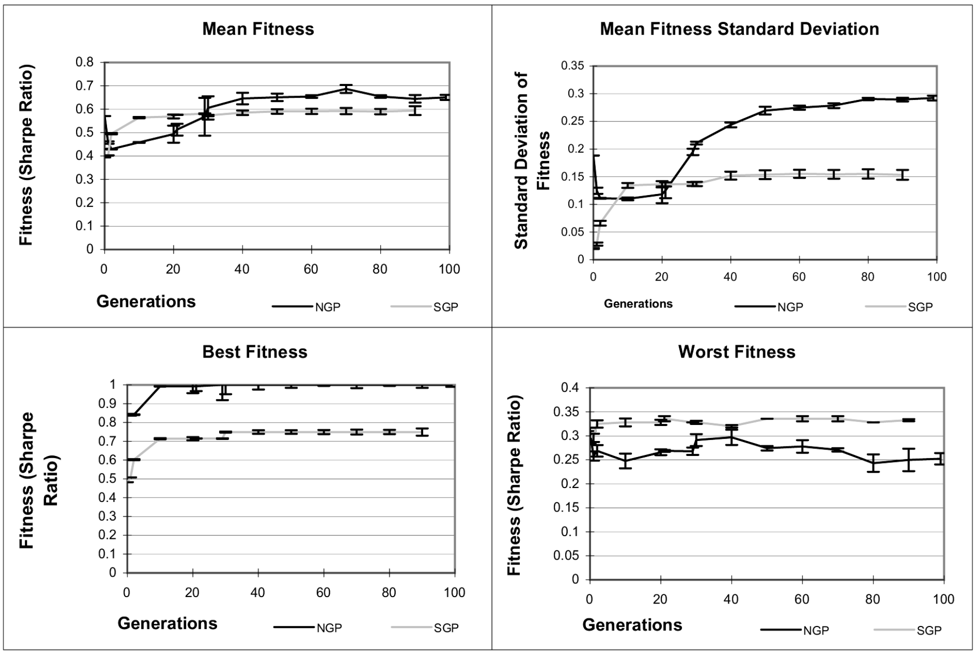


Figure 5.4: Retraining period-a population dynamics. The average value from 25 runs of each measurement was plotted for each generation ; the best and worst runs are indicated by error bars.

preservation techniques to affect the highest-fitness segment — this explains why the graph on the right of Fig. 5.3 tracks SGP during the first 20 generations. Student's T-test (less than 0.00002 for Generations 80 and 100) confirms that for the later generations, NGP provides a highly significant improvement.

Retraining population dynamics

The second research question posed was *"When retraining on data that comes from a different economic context, how quickly and how well does the diverse population adapt to a new environment?"*

Figs. 5.4, 5.5 and 5.6 illustrate how the populations for SGP and NGP evolve during retraining on data from different economic climates. In all three cases (Periods a, b and c), the standard deviation of fitness for NGP is significantly better for NGP than for SGP after about 20 generations. NGP always starts with a *best* fitness that is higher than SGP (indicating that the NGP individual is already better able to cope with the new economic climate than the SGP individual), and that superiority is retained as the population retrains. In fact, in all cases, NGP quickly finds the *ideal* Sharpe Ratio of 1.5 (giving the best adjusted fitness of 1.0) whereas SGP never finds that ideal individual.

When retraining, both SGP and NGP appear to converge on their respective *best* individuals at about the same rate, though SGP has the edge for Period c.

An interesting result from Figs. 5.4, 5.5 and 5.6 is the fact that the NGP standard deviation decreases for the first ten generations of retraining, then rises strongly. The initial generation

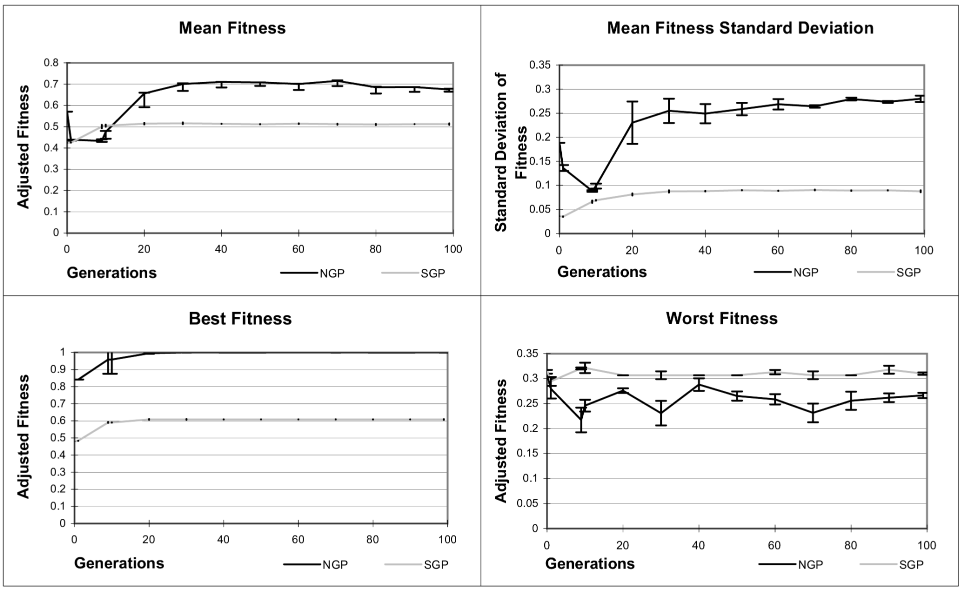


Figure 5.5: Retraining period-b population dynamics. The average value from 25 runs of each measurement was plotted for each generation ; the best and worst runs are indicated by error bars.

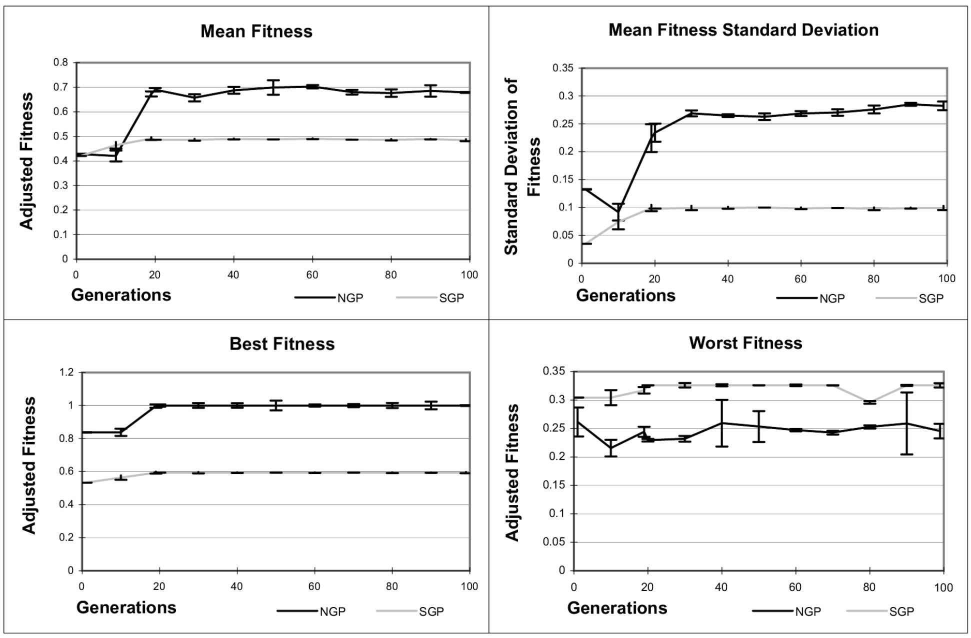


Figure 5.6: Retraining period-c population dynamics. The average value from 25 runs of each measurement was plotted for each generation ; the best and worst runs are indicated by error bars.

comprises individuals trained for a different economic context, and these individuals do not have a normal distribution of fitnesses in the context of the new data. In particular, (i) the fitness peak for the highest segment that is observed at the end of the original training is unlikely to occur in the context of the new data, and it is much more likely that a peak will occur in the middle segments, and (ii) the behavioural diversity of individuals in the highest segment is likely to be low. In the early generations, the NGP algorithm will discard many individuals whose behaviours are too well correlated with others, and replacements will be drawn from random samples — the overall effect will be to increase the number of individuals in the middle segments, thus producing a more peaked distribution. The NGP standard deviation therefore dips towards that of SGP, then rises strongly after the distribution has normalised, and the diversity-preserving algorithm has a stronger effect.

Out-of-sample individual robustness

The third research question posed was *"Are trained individuals resulting from the new system more robust when exposed to a new economic environment?"*

Case 1. The best individual trained from 31/07/1997 to 31/06/2002 (period a+b) and tested over the period 01/07/2002 to 31/12/2003 (period c).

Fig. 5.9 and Fig. 5.7 illustrate the performance robustness of NGP and SGP using robustness measures 1, 2 and 3 (see section 3.2 and 4.4) when the economic context changes.

1. Standard deviation of NGP and SGP is almost on the same level (a ranked T-test indicates no significant difference in the distribution). This indicates that NGP and SGP achieve a similar degree of robustness in terms of standard deviation of returns (measure 1).

2. However, NGP consistently generates considerably higher average returns than SGP. This indicates that NGP achieves a superior degree of robustness in terms of average returns (robustness measure 2).

3. SGP still outperforms the index portfolio. Although SGP and the index portfolio produce a similar level of returns, SGP has lower return volatility.

4. The Sharpe Ratio over the test period is also calculated and compared. The percentage reductions in Sharpe Ratio from in-sample training to out-of-sample validation of NGP is less significant than SGP. The reduction for NGP is 32% (T-test $t(30) = 8.41$, $p = 5.3 \times 10^{-11}$) and for SGP is 57% (T-test $t(30) = 9.90$, $p = 3.4 \times 10^{-13}$). NGP is more robust in terms of robustness measure 3.

Case 2. Retraining was performed on the last generation of individuals initially trained on Period a. After retraining on Period b (only), the best individual was tested on Period c.

The monthly ROI and monthly standard deviation of ROI in the new economic climate is plotted for both the best NGP individual and the best SGP individual. Retraining was

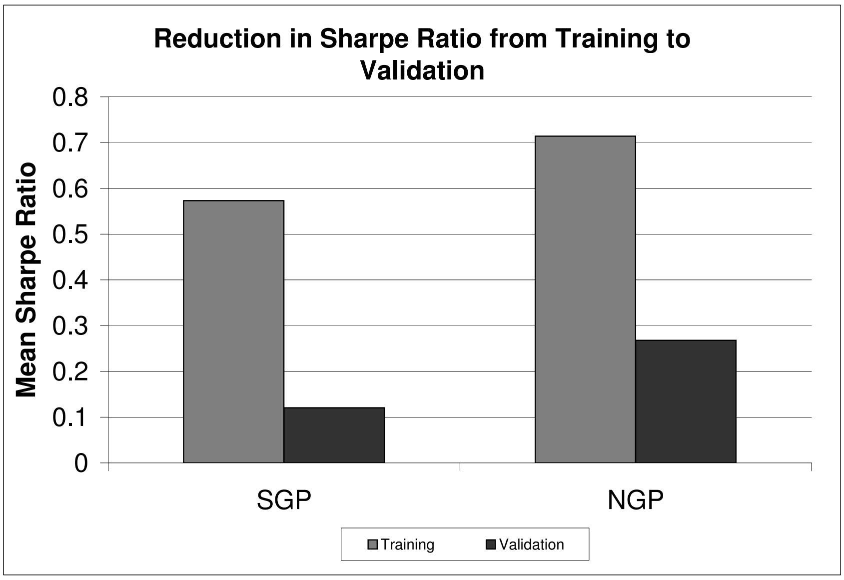


Figure 5.7: Robustness (Sharpe Ratio)

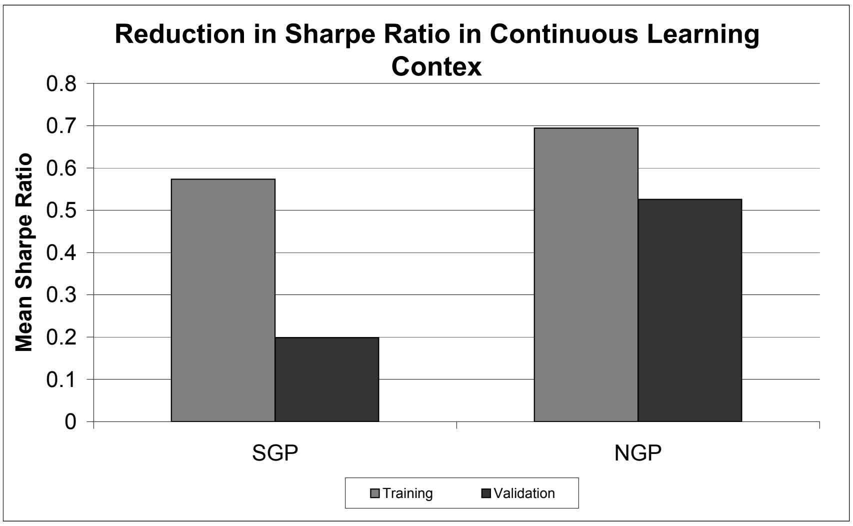


Figure 5.8: Retraining Robustness (Sharpe ratio)

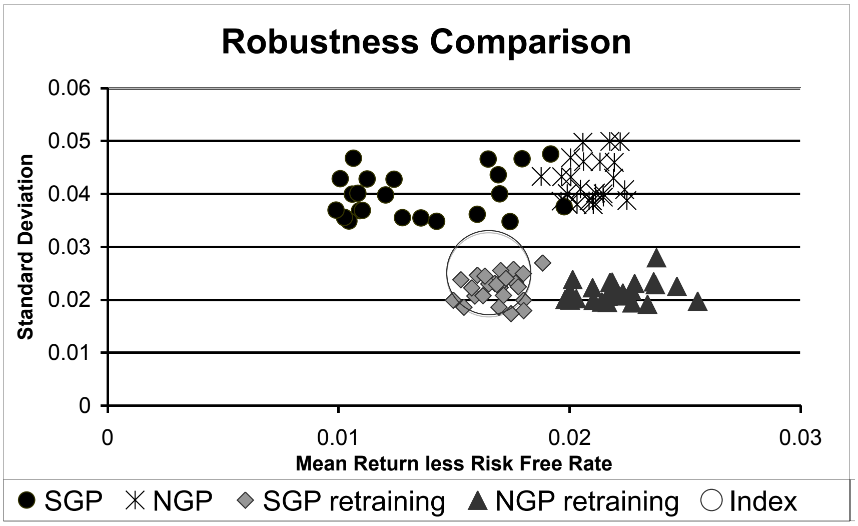


Figure 5.9: Robustness Comparison (Cases 1 & 2)

carried out on 18 months data and tested on the remaining 18 months data. This simulated a continuous-learning context where the GP system is periodically retrained to accommodate non-extreme changes in the economy.

1. At Fig. 5.9, the comparison of robustness plot indicates that NGP "retraining" in the continuous-learning context can produce more robust individuals than SGP "retraining" in terms of average return (measure 2). NGP retraining and SGP retraining share a similar level of volatility but the mean returns of NGP are clearly greater than those of SGP. This observation is consistent with Case 1. (robustness measure 2)

2. As in Case 1, the Sharpe ratios over the test period are calculated and compared. Fig. 5.8 shows that NGP (-24%, $t(30) = 7.28$, $p = 3.9 \times 10^{-9}$) suffers a much lower percentage reduction in Sharpe ratio from training to out-of-sample than SGP (-65%, $t(30) = 8.89$, $p = 6.3 \times 10^{-11}$). NGP is more robust in terms of degree of change in Sharpe ratio. (robustness measure 3)

Traditional training and validation vs continuous learning (Case 1 vs Case 2)

Fig. 5.9 also compares robustness of the trained individuals obtained through the traditional training and validation method and the continuous learning method. T-test comparison (Table 5.1) of return and standard deviation indicates a statistically significant difference between individual robustness evolved by systems with retraining and systems without retraining. Individuals evolved by NGP with retraining achieve a significantly lower standard deviation of ROI than those evolved by NGP using the traditional training and validation method, while maintaining a comparable level of return. Very similar observations can be applied when comparing individuals evolved by SGP with retraining against individuals evolved by SGP with traditional training and validation. Therefore, the continuous learning method generates more robust individuals according to robustness measure 1.

	Std. Devs.	**ROIs**
SGP retraining vs. SGP:	3.06×10^{-16}	0.28
NGP retraining vs. NGP:	3.24×10^{-16}	0.42

Table 5.1: Summary of Ranked T-test (p-values)

5.2 Multiple-scenario training GP

5.2.1 Experiment settings

The multiple-scenario approach is specifically designed for Type 3 environments where severe changes often occur and the fitness landscape before and after the change may display low degree of exploitable similarities. In this kind of environment, the GP learning process would NOT benefit in terms of robustness and efficiency from continuous adaptation (in other words using the previously trained individuals for continuous retraining). In this context, a 'two-step method' is used for this experiment. In the first step the GP system is trained to evolve individuals in one environment or one set of environments; and in the second step the trained individuals are validated directly in a previously unseen and profoundly different environment. The method does not involve an 'intermediate' step (which would involve retraining the previously trained individuals using a part of the data from the unseen environment).

The primary research question relating to the multiple scenarios approach is: *"are the best-of-run individuals from the two new systems more robust than the best-of-run individual from SGP when exposed to a volatile and previously unseen environment?"*

The experiment compares the performance of all three systems: SGP, MELG and MEVO.

Data

The training data for all three systems comprises time-series financial data for the 33 stocks taken from the period 31st January 1999 to 31st December 2004.

SGP and MELG use a training data set of financial time-series data taken from the period 31st January 1999 to 31st December 2004 (71 months).

For MELG (last generation only) and MEVO, the following three scenarios were chosen:

1. Bull market: 31/05/2003 to 31/12/2004 (19 months);
2. Bear market: 31/01/2000 to 31/05/2001 (16 months);
3. Volatile market: 31/01/1999 to 31/03/2000 (14 months).

Figure 5.10 shows the overall market index for Malaysian stocks, and a non-weighted portfolio index (constructed from all 33 stocks in which the simulator invests) [3], for the overall period under study; it also indicates the three scenario periods (bull, bear and volatile) and the validation period. The market and portfolio indices both show considerable volatility — the portfolio index is slightly more volatile than the overall market index, and so beneficial effects displayed by the GP system cannot be due solely to 'cherry-picking' of the least volatile stocks.

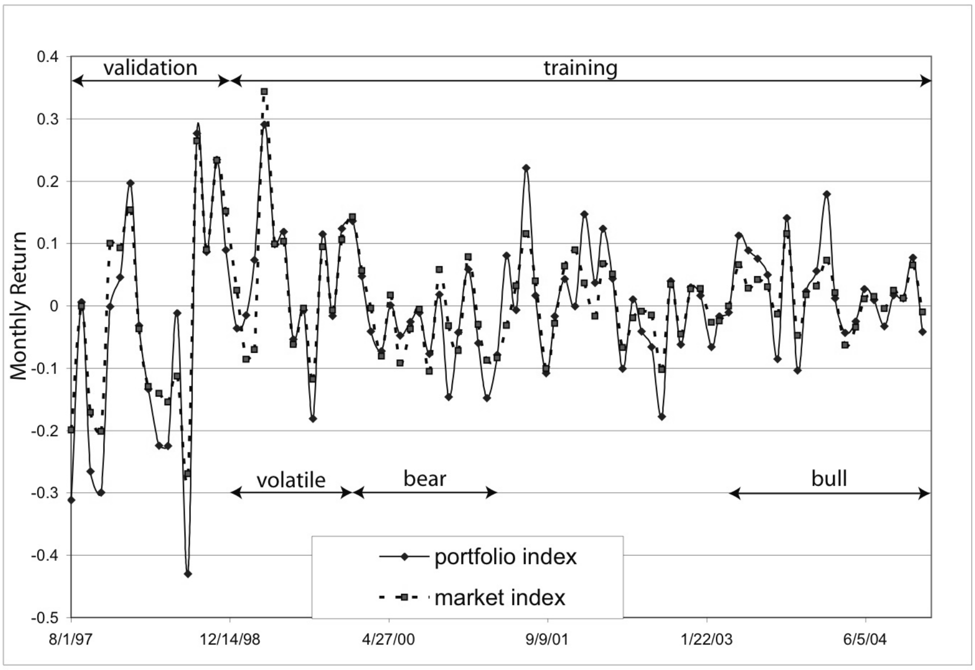


Figure 5.10: Market and portfolio indices (fractional monthly returns, 31st July 1997 to 31st December 2004) - scenarios and validation period.

Fitness

MELG and MEVO uses the Consistency Adjusted Fitness CaF as described in Equation 3.3 to calculate an individual (i)'s fitness.

$$CaF_i = \overline{F} \div \sigma \tag{5.2}$$

where $\overline{F}$ is the mean fitness of 3 scenario fitnesses (f_{bull}, f_{bear} and $f_{volatile}$, they are calculated using Equation 5.1) and σ is the standard deviation of f_{bull}, f_{bear} and $f_{volatile}$.

The fitness of canonical SGP is calculated using Equation 5.1).

Out-of-Sample Validation

All three systems are validated on a previously unseen 'out-of-sample' data set, comprising time-series financial data for the 33 stocks taken from the period 31st July 1997 to 31st December

[3]Because of the data availability, the number of stocks included in the investment universe is less than the number of total stocks listed in the index market. Therefore, the investment universe movement may not perfectly mirror the index movement. This is one of the reasons that the portfolio index are also observed. Another reason is that the portfolio index represents a buy-and-hold investment strategy which is often used as a benchmark for active investment strategies such as the hedge fund strategy used in the thesis.

1998. During this period, the Malaysia stock market was subject to great volatility including both the highest and lowest monthly returns in the entire period under study. From May 1998 to October 1998, the stock index lost more than 40%. Then, from November 1998 onwards, there was a remarkable performance from the market index, rising 23.3% in November alone.

This period has been deliberately chosen as a real test of robustness of individuals in a dynamic and hostile Type 3 environment. Episodes of very high volatility occur often in world stock markets and in emerging markets in particular. A successful hedge fund stock selection model must be robust — that is, it must be able to perform in both up and down markets.

For the out-of-sample validation, 25 complete training runs were performed (each run being seeded with a different random number) of each of the three systems (SGP, MELG, and MEVO) and the 'best-of-run' individual was selected from the final generation of each run.

The selected individuals were then validated on the previously unseen data and the results of the 25 runs are discussed in the following section.

5.2.2 Discussion of Results

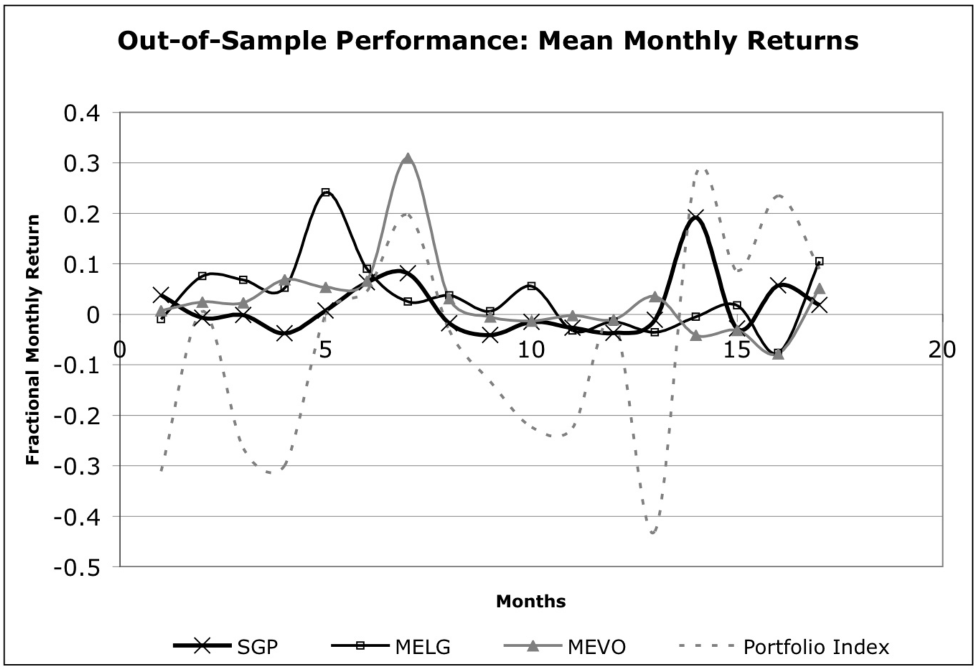


Figure 5.11: Mean monthly fractional returns.

Figure 5.11 shows the mean monthly returns (over 25 runs) on the validation data for all three systems (SGP, MELG and MEVO), together with the portfolio index returns. The index shows considerable volatility — it is more volatile than the overall market index seen in Figure 5.10, and so beneficial effects displayed by the new GP system cannot be due solely to 'cherry-picking' the least volatile stocks. SGP is not very volatile, but neither does it make much profit. Both MELG and MEVO appear to be adept at avoiding losses yet are still able to make good gains in positive months. Figure 5.12 show vividly the difference between the

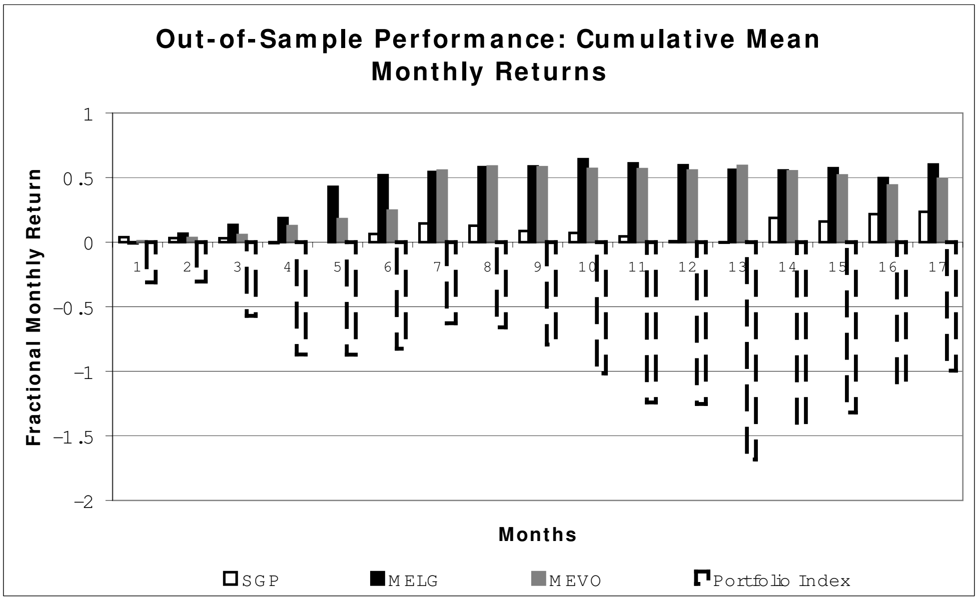


Figure 5.12: Cumulative monthly fractional returns.

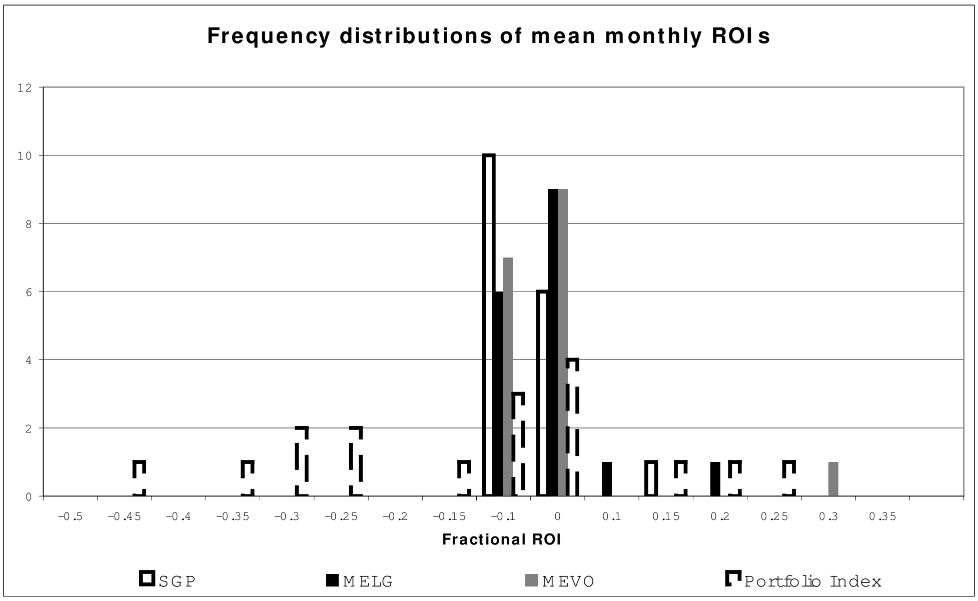


Figure 5.13: Frequency distributions of mean monthly fractional returns.

large cumulative losses of the portfolio index compared with the cumulative gains of MELG and MEVO.

Figure 5.13 gives another view of the mean monthly returns by plotting the frequency distributions of returns in the validation period. The portfolio index (shown dashed) is very volatile, whereas all three GP systems are much less volatile (though with significant positive fat tails).

Robustness

The performance of the three systems, using robustness measures 1 and 2, are illustrated in Figure 5.14, which shows standard deviation plotted against returns. Specifically, the figure plots returns in excess of the risk-free rate, and data has been added for the portfolio index and for a popular non-genetic technical strategy (using Moving Average Convergence Divergence (MACD) to select stocks). The portfolio index is shown to be not at all satisfactory, with both low returns and high standard deviation; the MACD approach performs much better than the portfolio index, but not as well as any of the three GP systems. In terms of robustness:

1. the three GP systems and MACD all have similar standard deviations (a ranked T-test indicates no significant difference in the GP distributions) and so by this measure no one system is more robust than another;

2. by contrast, the three GP systems and MACD differ in their returns while their standard deviations do not differ, so by measure 2 they are not equally robust. In order (from least to most robust) there is MACD, then SGP, the MEVO and finally (the best) MELG. This is further quantified below.

Fund managers use a very similar approach to the robustness measures 1 and 2 — they use the Sharpe Ratio [143] which determines the ROI (in excess of the risk-free rate) per unit of risk (given by the standard deviation). Since the standard deviations in this case are the same, a Sharpe Ratio comparison also provides a quantitative comparison of returns and thus of the robustness measure 2. Therefore, the Sharpe Ratios (across 25 runs) have been calculated for each of the three GP systems.

Comparison of the Sharpe Ratio distributions shows that all three systems achieve substantially better results than the portfolio index (as expected from Figure 5.14) and a ranked T-test comparison of the Sharpe Ratios indicates a statistically significant difference between all three systems. The p-values (the probabilities that two compared distributions are from the same population) are presented in Table 5.2. The means of the Sharpe Ratio distributions are: 0.125 (SGP), 0.305 (MEVO), and 0.421 (MELG) — MELG is substantially the most robust system of the three.

Compare SGP with MELG:	4.01×10^{-16}
Compare SGP with MEVO	4.13×10^{-16}
Compare MEVO with MELG	4.62×10^{-9}

Table 5.2: Summary of Ranked T-test (p-values)

The third measure of robustness selected determines how much the mean return per unit of risk reduces when moving from the training set to the validation set. This is shown in Figure 5.15. The percentage reductions in Sharpe Ratio (and associated p-values from a ranked

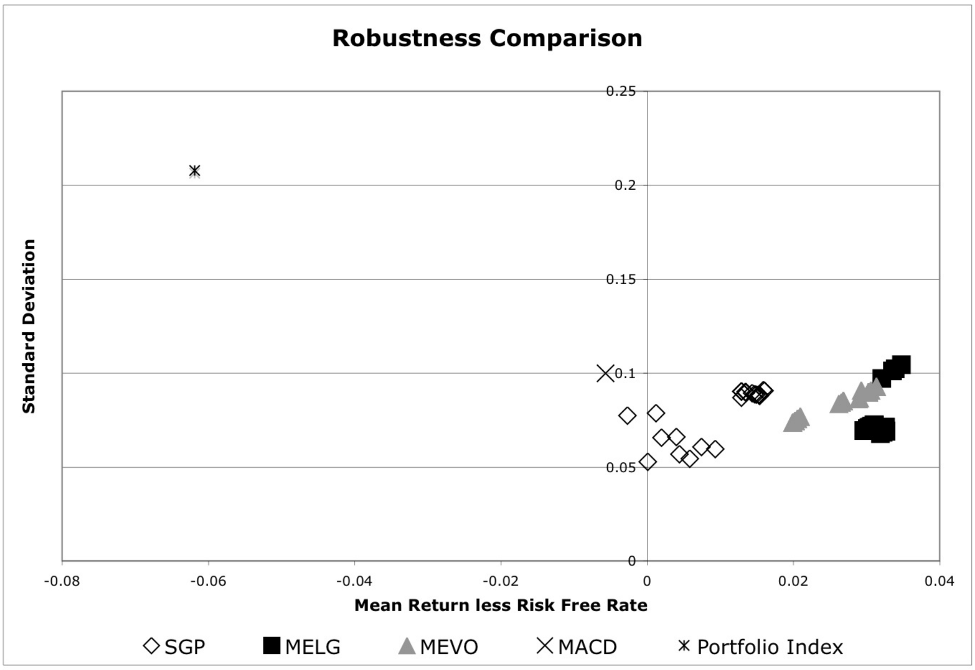


Figure 5.14: Robustness comparison.

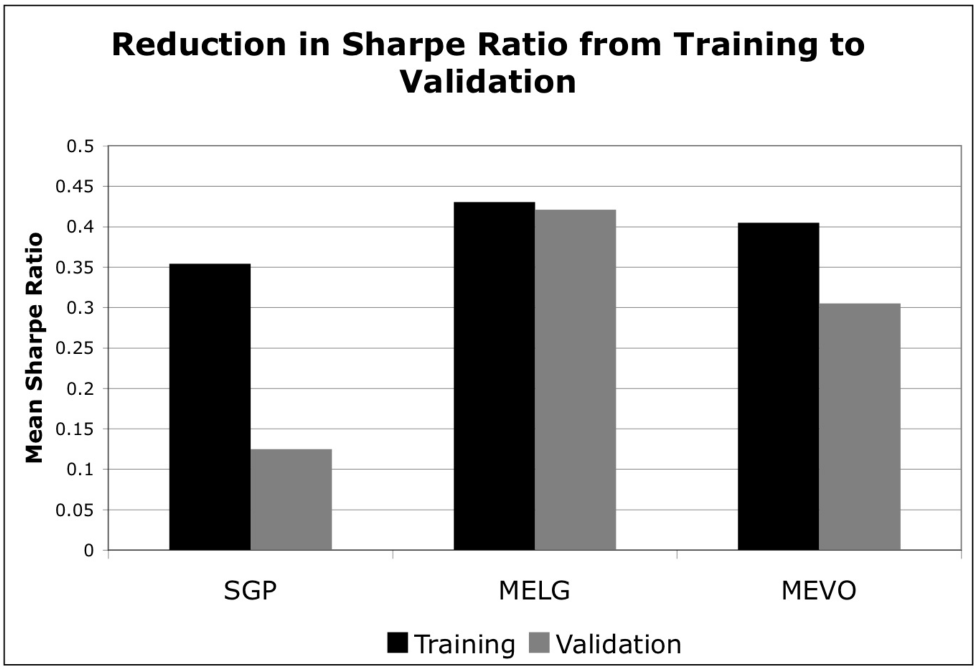


Figure 5.15: Robustness measure: drop in Sharpe Ratio.

T-test) were 65% for SGP (1.4×10^{-11}) , 25% for MEVO (3.1×10^{-8}) and just over 2% for MELG (0.92), indicating a substantial robustness advantage for MELG.

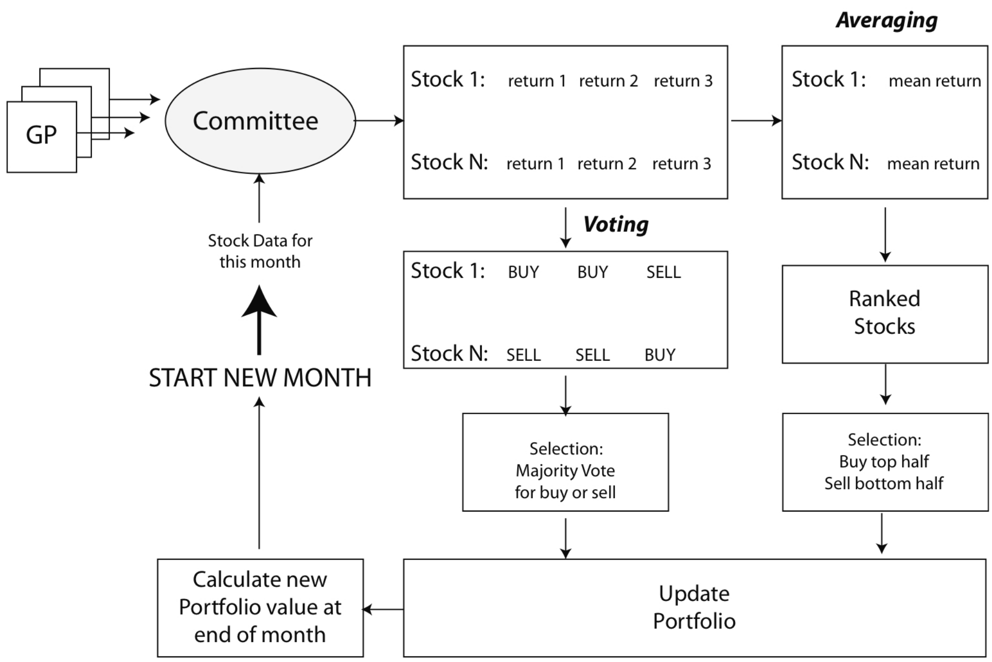


Figure 5.16: The Committee in action: either Averaging or Voting.

5.3 Committee Voting GP

5.3.1 Experiment Settings

The committee is implemented as part of the investment simulator. The simulator is used both during GP evolution (where it is called by the fitness function) and during validation, but the committee is only used during validation.

There are two possible combining mechanisms, either *Averaging* or *Voting*, as shown in Figure 5.16.

Averaging: The first mechanism averages the team members' output. This results in a mean predicted return for the next 30 days for each stock. The stocks are then ranked in order of this mean predicted return, with the top half being selected for buying and the bottom half being selected for selling.

Voting: With the second mechanism, each team member uses its predicted returns to generate its own ranking of all the stocks; this is then converted into a buy decision for those stocks in the top half of the ranking and a sell decision for those stocks in the bottom half.

After the 'buy'̀sell' recommendations have been calculated for all team members and for all stocks, a majority voting method is applied to each stock and a final buy or sell decision is derived for that stock. With majority voting, if a stock has more 'buy' recommendations than 'sell' recommendations, it will be bought - otherwise it is sold.

Because in this application GP generates the factor models for ranking purpose, the experiment adopts the voting mechanism.

The primary research question concerning the committee voting approach is: *"does a voting system provide more robust results than the best-of-run individual from SGP when exposed to a volatile and previously unseen environment?"*

This experiment compares the performance of an SGP individual with the Voting system comprising three 'best-of-run' individuals derived from three GP evolutions with different training data sets.

Fitness

The fitness f for an individual is given by Equation 5.1.

Data

This experiment uses the same data as the multi-scenario GP to train and validate the algorithms.

The committee

During validation, the 'Voting investment' simulator is augmented with a committee structure containing a team of three individuals.

In investment portfolio optimisation the trading is carried out monthly and the aim is to pick those stocks that will perform well *regardless* of whether the market in the following month will be bull, bear, or volatile. Thus, this experiment does not follow the otherwise obvious strategy of detecting the current market conditions and using an individual that has been trained only on that one market condition. Rather, the voting team comprises the 'best-of-run' individual chosen from each of the final populations of three GP systems, each of which has been trained on only one market condition, — in other words, the three systems have undergone separate training with predefined distinctively different training data sets representing the three market environments - 'bull', 'bear' and 'volatile'. The expectation is that the behavioural correlation between team members will be low.

5.3.2 Discussion of Results

Figure 5.17 plots the frequency distributions of returns in the validation period. The market and portfolio indices (dashed lines) are both very volatile; SGP makes a fairly consistent slight loss balanced by some gains in a positive short fat tail. The Voting system makes a fairly consistent slight gain but with a short fat positive tail.

Robustness

In this experiment, as in the previous experiment, robustness measures 1, 2 and 3 are observed. In each case there were 25 training runs, each run being seeded with a different random number: for SGP, the reported mean ROI and standard deviation are the results of applying the best individual from the final generation to the validation data; for the Voting system, a voting pool of 3 individuals was selected from final generations of each of the 25 runs — the voting pools were then applied to the validation data and the mean ROI and standard deviation calculated.

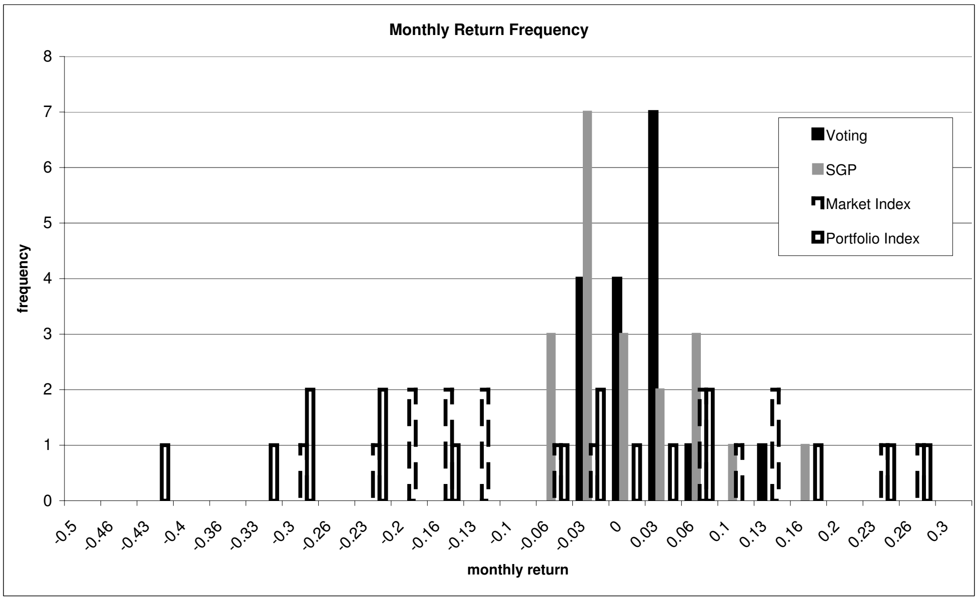


Figure 5.17: Frequency distributions of mean monthly fractional returns.

The performance of the two systems, using robustness measures 1 and 2, are illustrated in Figure 5.18.

- The Voting system and SGP system have similar standard deviations and so by this measure no one system is more robust than another.

- By contrast, the Voting system is superior in terms of return. Figure 5.19 gives the frequency distribution for the Sharpe ratio for both SGP and the Voting system. As above, in each case there were 25 training runs, each run being seeded with a different random number. Both systems beat the portfolio index Sharpe Ratio of -0.297 (a negative return on investment!), but the Voting system is substantially superior. A ranked T-test result is displayed in Table 6.3.4 and indicates a convincing difference between the two systems. Therefore, the Voting system is more robust than SGP in terms of measure 2.

The percentage reductions in Sharpe Ratio (and associated p-values from a ranked T-test) were 65% for SGP (1.4×10^{-11}), and 13% for the Voting system (5.6×10^{-4}). The performance of the voting system in the different environments is more resilient to changes, thus more robust.

SGP vs. Voting (Mean ROI):	3.8×10^{-8}
SGP vs. Voting (Standard Deviation):	4.0×10^{-4}
SGP vs. Voting (Sharpe Ratio):	4.32×10^{-16}

Table 5.3: Ranked T-test results (p-value)

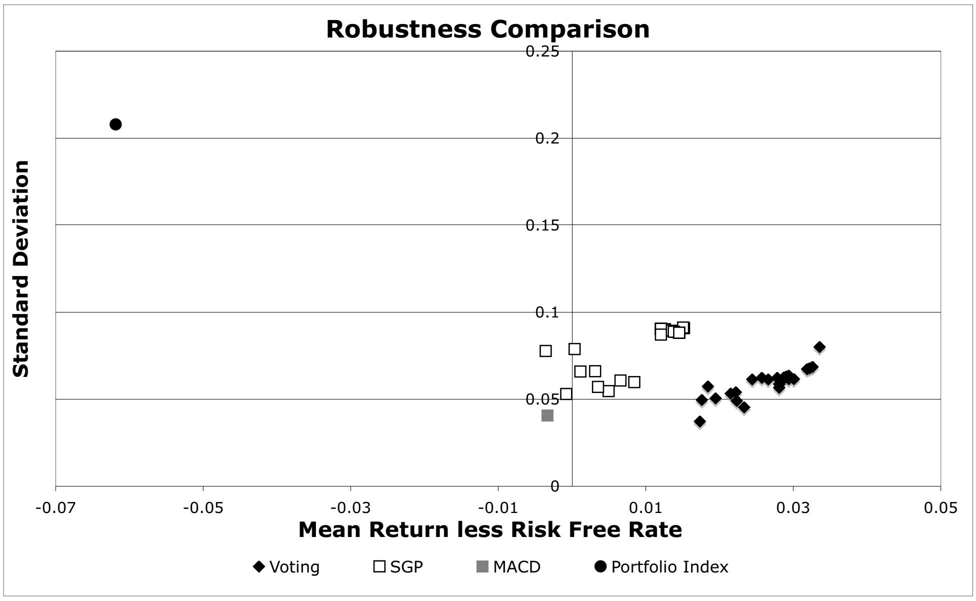


Figure 5.18: Robustness comparison.

5.3.3 Multiple Scenarios Approach vs. Voting Approach

Robustness of the two approaches (MELG vs. MEVO vs. voting system) is compared in Figures 5.20 and 5.21. Figure 5.20 compares the robustness of the three systems in terms of performance in relation to associated risk (robustness measures 1 & 2). Figure 5.21 compares the robustness of the three systems in relation to change of the performance in different environments (robustness measure 3). In Figure 5.20, MEVO is slightly outperformed by MELG and the Voting system as, given the same level of risk, it does not yield higher returns than the other two systems. MELG gives higher returns more consistently than the voting system, at the cost of higher risk. The same pattern emerges in Figure 5.21; again, MELG has the best performance, beating MEVO, and SGP has the worst performance.

5.3.4 Summary

Experiment results were presented based on simulation of a portfolio system which used a GP system to evolve a non-linear factor model for stock-picking, coupled with an Investment Simulator that modelled a long-short, market-neutral, sector-neutral hedge fund, trading Contracts for Difference (CFDs) in the highly volatile Malaysia stock market. Historical stock data (both technical and fundamental) was used from the period 1997-2004.

In summary, the evaluation provided in this chapter produced positive results for all of the three new algorithms proposed and the results demonstrated that these approaches are viable strategies for improving robustness of solutions evolved by GP in dynamic environments.

- Behavioural diversity preservation GP

  The main objective of this experiment was examination of GP behaviour in a continuous learning context that was most suitable for a Type 2 environment which had a medium

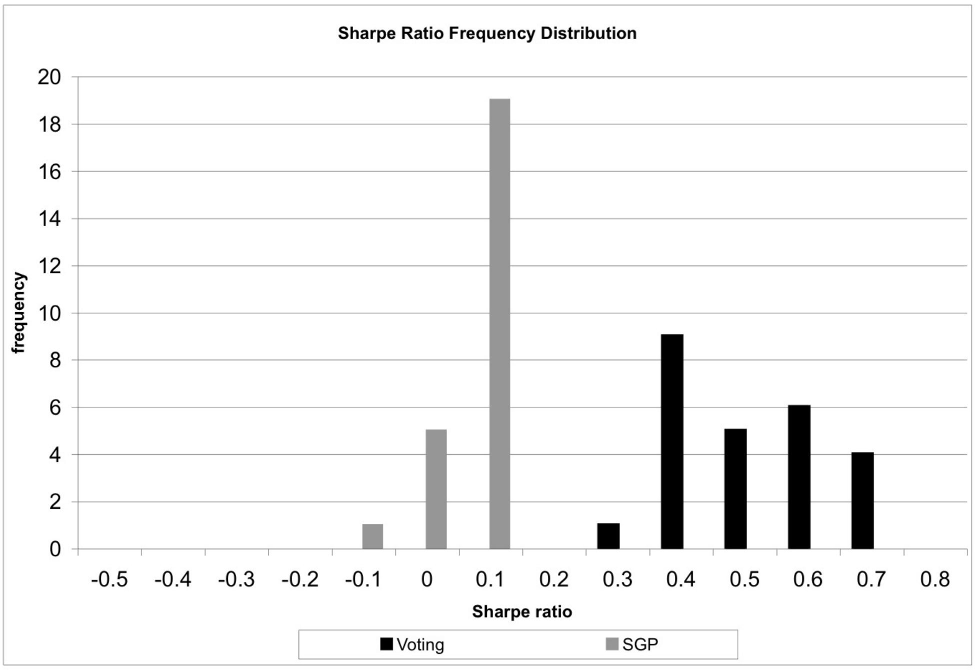


Figure 5.19: Frequency distributions of Sharpe ratios. Voting average = 0.567292236, $\sigma = 0.111250913$, SGP average = 0.12483279, $\sigma = 0.062651544$.

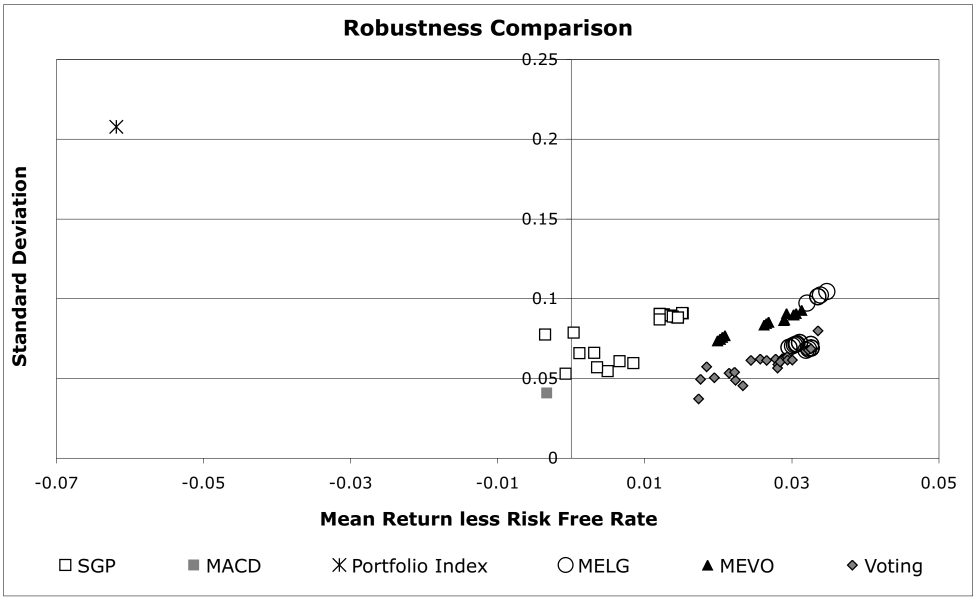


Figure 5.20: Robustness comparison of all three systems.

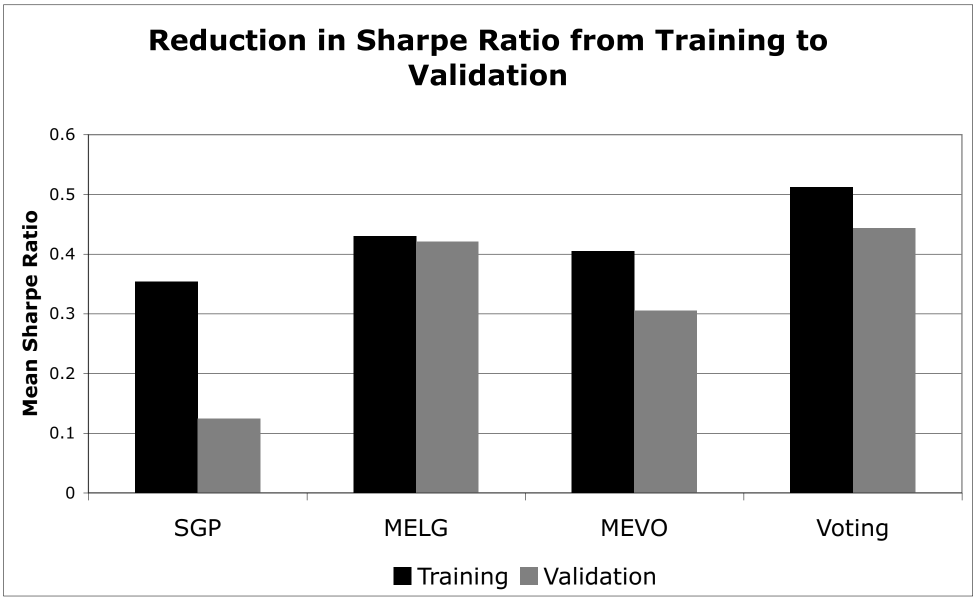


Figure 5.21: Drop in mean Sharpe Ratios for all three systems.

level of dynamics. The algorithm was tested at three different levels. The first level was population diversity, the second, population dynamics during retaining and the third, individual robustness in continuous learning.

Statistical analysis demonstrated that the new algorithm increased the population diversity of phenotypic behaviour and positively affected the global population dynamics in terms of fitness and adaptivity. It also demonstrated that the new algorithm consistently evolved more robust individuals in terms of robustness measures 2 and 3 than the canonical GP.

- Multiple-scenario GP

  The main objective of this experiment was the examination on the robustness of MELG- and MEVO-evolved individuals in a Type 3 environment. Statistical analysis of the results overwhelmingly demonstrated that MELG and MEVO evolved more robust individuals than canonical GP, using robustness measures 2 and 3 in extreme dynamic environments. There were no significant differences using robustness measure 1.

- Committee-voting GP

  The main objective of this experiment was examination of the robustness of individuals evolved in a Type 3 environment. The statistical analysis demonstrated that the voting system was superior in terms of robustness measures 2 and 3 in extreme dynamic environments compared with the canonical GP.

  When comparing the committee-voting GP with MELG and MEVO systems, the voting GP was slightly outperformed by the two multiple-scenario GP algorithms in terms of robustness measures 2 and 3.

The excellent profits produced in the experiments over unseen data provide a justification for further investigation. For example, is there anything intrinsically better about a GP system when compared with, for example, a machine-learning technique such as Support Vector Machine (SVM)? In the next chapter, a head-to-head evaluation is provided of GP and SVM applied to the Malaysia case study.

Chapter 6

Comparing Techniques for Financial Portfolio Optimisation -GP vs Support Vector Machine

The evaluation of the 'multiple-scenario' algorithm, the 'committee voting' algorithm and the 'behavioural diversity preservation' algorithm in the previous chapter demonstrated that they each enhance robustness of GP solutions in the Malaysian test case. Stock selection for hedge funds has previously been tackled by SVMs as reviewed in Section 2.4. But which is the better? The aim of this chapter is to provide a head-to-head analysis and comparison of GP and SVM and to show that the GP approach is qualitatively different to SVM, and produces much higher profits as a result for this real-world problem. This chapter has been published in [178].

The experiments included both a comparison of Return On Investment (ROI) and a comparison of the two techniques when extended with a 'voting' mechanism to improve both ROI and robustness to volatile markets. Robustness is an important additional dimension to this comparison, since the markets (the environment in which the GP or SVM solution must survive) are dynamic, unpredictable and unforgiving.

In addition to GP, there are two other major automatic regression techniques, namely, Artificial Neural Networks (ANNs) and SVMs. However, SVMs, the most popular kernel machine [36], based on recent advances in statistical learning theory has become increasingly important rapidly replacing ANNs which using traditional linear multi-layer regression approaches [14]. The main advantages of SVMs over linear learning machines such as ANNs are [82, 140, 144, 14]: they are less prone to overfitting; they need less parameters to tune; they have more solid mathematical theoretical foundation and they are flexible through an appropriate choice of kernel to generate or to replace many existing machine learning architectures e.g. feedforward neural networks, radial basis function networks.

This chapter provides an explanation of the portfolio optimisation systems, followed by detailed comparisons of GP and SVM in terms of (i) returns on investment and (ii) extension with a voting system; the results are discussed and an explanation for GP's superiority is proposed. Finally, a further experiment is conducted to investigate the proposed explanation.

6.1 The Portfolio Optimisation Systems

6.1.1 The SGP System

The selected test system comprises a standard GP system (SGP) coupled with an investment simulator. The SGP system is identical to the system described in Sections 4.2 and (with voting) in Section 5.1.3

6.1.2 The SVM System

The selected SVM system supports a market-neutral hedge fund simulation identical to that used in the SGP system. It consists of a support vector regression system (SVR) and an investment simulator. The SVM system has been implemented independently by Martin Sewell, the co-author of the paper mentioned at the beginning of the chapter[178].

The resulting SVM is used in the Investment Simulator during the validation phase; however, unlike SGP, the Investment Simulator is not used during training. Instead, during training the SVR is provided with the 1-month future prices of each stock. The SVR system outputs stock return predictions by aiming to fit the 1-month future prices, and regresses to a nonlinear equation that should be very similar to that evolved by SGP.

For the forecasting problem of a univariate time series, the inputs of SVR are the past, lagged observations of the data series and the outputs are the future values. Each set of input patterns is composed of any moving window of fixed length within the data series. The mapping function of the form can be described as below:

$$y_{t+1} = f(y_t, y_{t-1}, ..., y_{t-p}) \tag{6.1}$$

Here, y_t means the observation of price return at time t; and p means the dimension of the input vector or the number of the past observations related to the future value.

Performance Criteria

The SVM prediction performance is evaluated using the mean square error (MSE) — the measure of the deviation between the actual and predicted values. The smaller the values of MSE, the closer the predicted time series values to the actual values.

$$MSE = \frac{\sum_{i=1}^{n}(a_i - p_i)^2}{n} \tag{6.2}$$

where a_i and p_i are the actual values and predicted values.

6.2 Profitability Comparison

6.2.1 Experiment

The research question posed is: *"does SGP produce more profitable results than SVM when exposed to a volatile and previously unseen environment?"*

The experiment compares the returns on investment of an SGP individual with the SVM system. The basic SGP and SVM parameter settings are given in Table 6.1 and Table 6.2.

Population size (N)	1000
Method of generation	Ramped half and half
Function set	{+, -, *, /, Exp}
Terminal set	18 firm-specific factors
Selection scheme	Fitness proportionate
Criterion of fitness	Monthly Sharpe ratio
Elitism	10 (1%)
Crossover	950 (95%)
Mutation	40 (4%))
Termination criterion	100-generation evolution
Initial Max. depth	6

Table 6.1: SGP Parameter Settings

Kernel	**Gaussian radial basis function**
C (margin of tolerance of error)	0.01
ϵ **(data point accuracy)**	0.1
γ **(width of sampling Gaussian)**	1

Table 6.2: SVM Parameter Settings

6.2.2 Data

All systems use an Investment Simulator that has an investment universe of 33 Malaysian stocks. The training data for both systems comprises time-series financial data from 31st January 1999 to 31st December 2004 (71 months).

Because SVM needs a cross-validation procedure during the training to determine kernel parameters, SVM's training data is split into two phases:

1. Initial training period: 31 Jan. 1999 to 31 Dec. 2002
2. Cross-validation period: 31 Jan. 2002 to 31 Dec. 2004

6.2.3 Out-of-Sample Validation

The two systems are validated on a previously unseen 'out-of-sample' data set, comprising time-series financial data for the 33 stocks taken from the period 31st July 1997 to 31st December 1998.

- For SGP out-of-sample validation, the 'best-of-run' individual was selected from the final generation and 25 runs were recorded.
- For SVM out-of-sample validation, the trained SVM was tested on the same investment simulator for SGP and since SVM is deterministic only one run is recorded. This was carried out independently by Martin Sewell.

6.2.4 Results and Discussion

Table 6.2.4 compares the average monthly ROI of 25 SGP runs against SVM. For comparison data for the portfolio index has been added and a common technical strategy used by investors; Moving Average Convergence Divergence (MACD). It shows clearly that SGP (1%) can produce more profitable results than SVM (-0.4%). SVM is significantly worse than SGP (about three standard deviations away from the SGP mean), and even suffered a greater loss than MACD.

SGP	0.0096 ($\sigma = 0.005$)
MACD	-0.0033
SVM	-0.0047
Portfolio Index	-0.06

Table 6.3: Comparing Returns on Investment (ROI)

6.3 Impact of Voting Mechanism

A voting mechanism can be used to improve both ROI and robustness in volatile markets [176], so the next step is to incorporate voting into both SGP and SVM.

6.3.1 The SGP Voting System (SGP-V)

During validation, the investment simulator is augmented with a committee containing a team of three individuals which have been trained in three different environments.

6.3.2 The SVM Voting system (SVM-V)

In order to be consistent with the SGP-voting approach, SVM-V also incorporates the voting committee in the investment simulator during validation. The SVM-V voting committee consists of three entirely distinctive SVMs trained on specialised scenarios such as ‘bull’, ‘bear’ and ‘volatile’.

6.3.3 Experiment

The research question posed is: *“does an SVM voting system provide more robust results than a SGP voting system when exposed to a volatile and previously unseen environment?”* The experiment devised compares the performance of 4 systems: SGP, SGP-V SVM, and SVM-V.

Data

For the three special-scenario GP evolutions, the following three market contexts were chosen:

1. Bull market: 31 May 2003 to 31 Dec 2004 (19 months);

2. Bear market: 31 Jan 2000 to 31 May 2001 (16 months);

3. Volatile market: 31 Jan 1999 to 31 Mar 2000 (14 months).

The above three market contexts were also chosen to train and cross-validate the three distinctive SVMs specialising in the bull, bear and volatile market conditions. However, in order to increase the number of available data points and thereby obtain improved performance from the SVMs, each market context data set was augmented with a certain amount of data from the 'normal' market context period from 1st June 2001 to 30th May 2003. In all cases, two thirds of the context data was used for training and one third for cross-validation.

6.3.4 Results and Discussion

Table 6.3.4 compares ROIs of the four systems, SGP, SGP-V, SVM and SVM-V. Both GP systems outperform the two SVMs: SVM-V is significantly worse than SGP-V (about three standard deviations away from the SGP-V mean). It is to be noted that integrating a voting mechanism can remarkably enhance profitability of both GP systems and SVM systems; the improvement in terms of pure profitability for the GP system is over 188% and for the SVM system is 240%. So what does this say about "robustness", and how can it be measured?

SGP-V	0.026 ($\sigma = 0.006$)
SGP	0.0096 ($\sigma = 0.005$)
SVM-V	0.0077
MACD	-0.0033
SVM	-0.0047
Portfolio Index	-0.06

Table 6.4: Voting Improves Returns on Investment

Simplistically, robustness might be taken to be synonymous with 'low variance' — that is to say, the performance of the individual does not alter much, despite the extreme volatility of the market environment. However, in practice there is a much more exacting requirement: it is not helpful to an investor to know that an individual robustly (with low variance) makes a loss regardless of the market! A much more helpful measure is to know that the individual combines two qualities of (i) high return on investment and (ii) low variance in the face of extreme volatility.

The performance of the four selected systems, using robustness measures 1 and 2 above, are illustrated in Figure 6.1, which shows standard deviation plotted against returns in excess of the risk-free rate. The portfolio index is shown to be not at all satisfactory, with both low returns and high standard deviation; the MACD approach performs much better than the portfolio index, but not as well as any of the two GP systems. In terms of robustness:

1. the two GP systems have similar standard deviations (a ranked T-test indicates no significant difference in the GP distributions), and so by measure 1 no one system is more robust than another. By contrast, the SGP and the SGP-V systems differ greatly in their returns. The SGP-V system consistently produces superior returns than SGP. Thus, by robustness measure 2, the SGP-V system is more robust than SGP;

2. sharing a similar trend, the standard deviations of SVM and SVM-V do not differ significantly, but SVM-V greatly improves its return. Again, by robustness measure 2, the

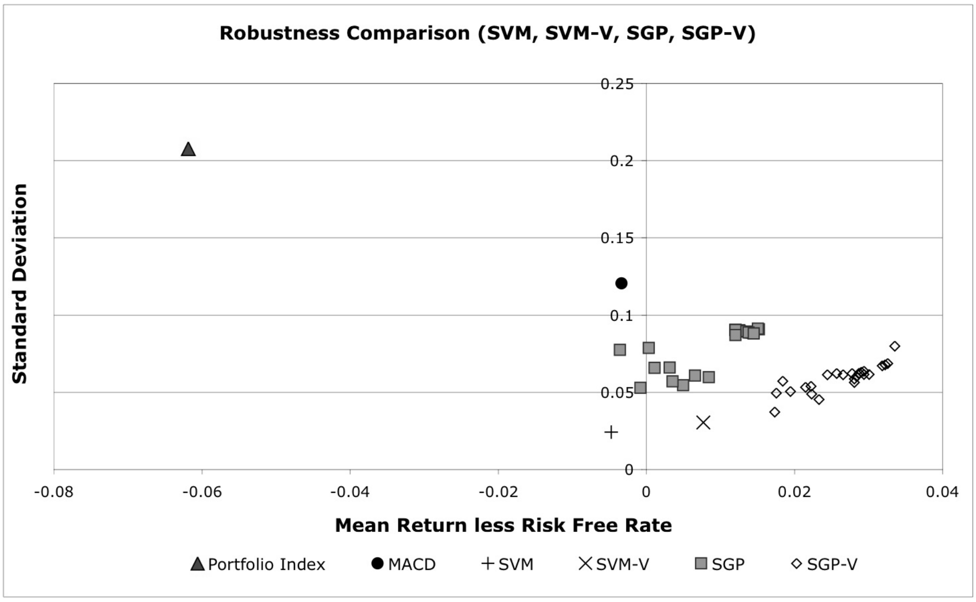


Figure 6.1: Robustness comparison.

SVM-V is more robust than SVM.

The Sharpe Ratios for the two SMV systems and the two GP systems (averaged across 25 runs) have been calculated — see Table 6.3.4. SVM-V is over five standard deviations worse than the SGP-V mean, and SVM is over ten standard deviations worse than SGP. For the two GP systems, comparison of the Sharpe Ratio distributions shows that the two systems achieve substantially better results than the portfolio index (as expected from Figure 6.1) and a (non-parametric) Ranked T-test comparison of the Sharpe Ratios indicates a statistically significant difference between the two systems. The p-values (the probabilities that two compared distributions are from the same population) are presented in Table 6.3.4.

SGP-V	0.44 ($\sigma = 0.039$)
SGP	0.11 ($\sigma = 0.033$)
SVM-V	0.22
MACD	-0.03
SVM	-0.196
Portfolio Index	-0.297

Table 6.5: Sharpe Ratio Comparison

For the two SVM systems, the comparison of the Sharpe Ratio clearly shows that the SVM-V achieves a significantly higher Sharpe ratio.

The combined results of the four systems overwhelmingly indicate that 1) SGP-V is more robust than SVM-V and 2) voting can enhance both GP and SVM robustness.

SGP vs. SGP-V (Mean ROI):	3.8×10^{-8}
SGP vs. SGP-V (Standard Deviation):	4.0×10^{-4}
SGP vs. SGP-V (Sharpe Ratio):	4.32×10^{-16}

Table 6.6: Ranked T-test results (p-value)

6.4 Learning to optimise profits vs predicting returns

Having determined that SGP is better than SVM, and SGP-V is better than SVM-V, for this portfolio optimisation problem, the obvious question is: 'why?' According to the literature, SVMs are very good at nonlinear regression, and therefore should be good at predicting stock returns. GPs are also known to be good at symbolic regression, but why are they so superior in solving this particular problem?

It is conjected that the SGP (and SGP-V) system is performing a different task to SVM (and SVM-V) — rather than predicting the returns that each stock will provide in the next month, it provides a stock ranking that optimises the overall performance of the investment simulator.

Consider the fitness function in the GP system: it takes an individual equation and passes it to the investment simulator, which applies the equation to every stock to give a number which is used to rank the stocks in order from best to worst; this is repeated with that same equation for every month in the simulation and the Sharpe Ratio of the investment simulator's performance is returned. This Sharpe Ratio is the basis for the fitness value. Thus, each GP individual is given a fitness value based not on how accurately it predicts the returns of stocks, but on how well it ranks stocks such that the simulator optimises its Sharpe Ratio.

The GP systems do not just predict returns — they optimise excess profit per unit-risk. They can do this because the investment simulator is embedded in the fitness function — something that is not at all easy for an SVM to achieve.

The proposed hypothesis is that learning to optimise profits is better than predicting returns, but how can this be checked? The obvious way is to assess the performance of a GP system that only predicts profits and does not interact with the investment simulator during training.

6.4.1 Prediction GP (pSGP)

In order to replicate the prediction approach adopted by SVM in a GP context, pSGP (and pSGP with voting: pSGP-V) detaches the investment simulator from the evolution process and only employs the simulator for validation. Thus, individual fitness f is measured purely as accuracy in predicting returns of n stocks, using the mean squared error as in the SVM system, as follows:

$$f = \frac{1}{n} \sum_{i=1}^{n} (R_{i_{prediction}} - R_{i_{actual}})^2 \qquad (6.3)$$

$R_{i_{prediction}}$ = predicted price return of the stock i

$R_{i_{actual}}$ = actual price return of the stock i

6.4.2 Experiment

The null hypothesis is that pSGP (pSGP-V) will continue to perform as well as SGP (SGP-V), and therefore the proposed hypothesis (that GP is performing a substantially different task to SVM as a result of the embedding of the investment simulator) is incorrect. Obviously, it is hoped that the null hypothesis shall be proven false!

The following experiment compares the performance of all six systems: SGP, SGP-V, pSGP, pSGP-V, SVM and SVM-V. As before, both ROI and robustness are compared.

Comparison of Returns

Figure 6.2 compares ROIs of all six systems, SGP-V, SGP, SVM-V, SVM, pSGP-V and pSGP. The two GP systems that only use returns prediction (pSGP and pSGP-V) perform substantially worse than both SGP and SGP-V, with ROIs that fall between those of SVM and SVM-V.

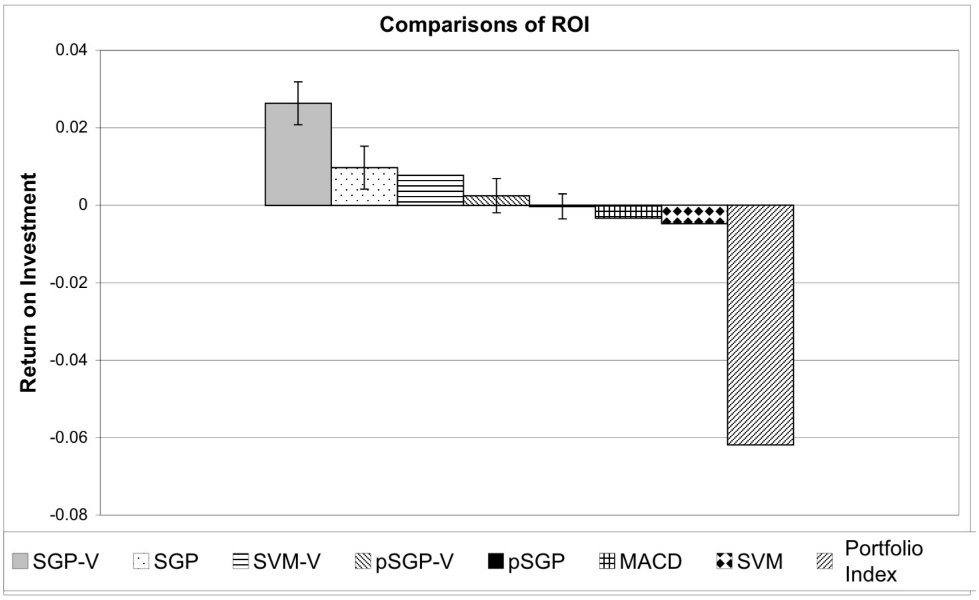


Figure 6.2: ROI comparison.

Robustness Comparison

The performance of the 6 systems (pSGP, pSGP-V, SGP, SGP-V, SVM, SVM-V), using robustness measures 1 & 2, are illustrated in Figure 6.3, which shows standard deviation plotted against returns. The 6 systems have also been compared with a random strategy which gives random stock rankings for the investment simulator.

- pSGP has similar risk but very different returns to SGP (a Ranked T-test gives a p-value of 5.88×10^{-9}), and in fact is closer to SVM's returns (both pSGP and SVM generate negative returns).

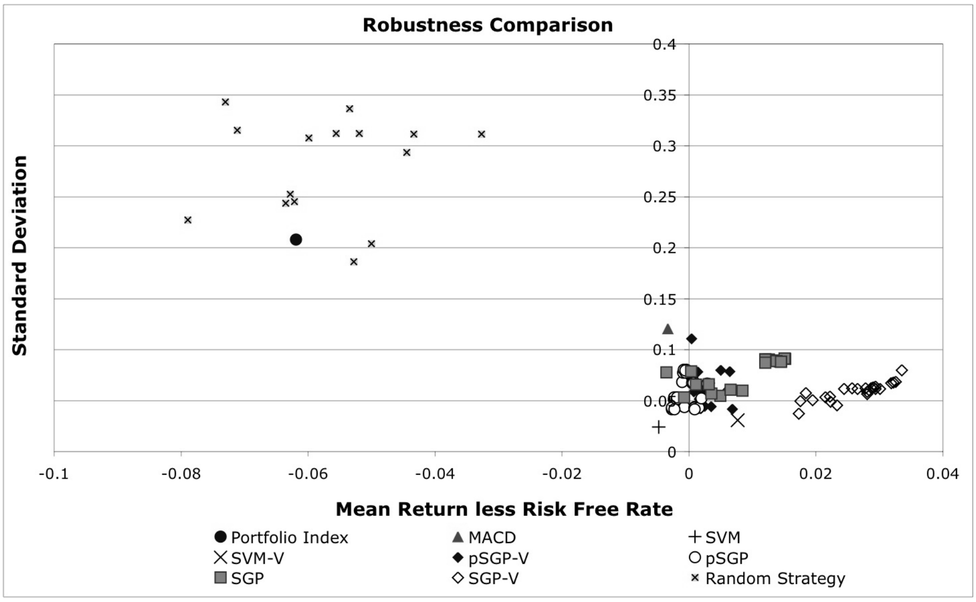


Figure 6.3: Robustness comparison.

- Similarly, pSGP-V has similar risk but very different returns to SGP-V (a Ranked T-test gives a p-value of 4.52×10^{-16}), and is closer to SVM-V's returns.

The above results demonstrate that the null hypothesis is false, and give confidence in the proposed assertion that the GP systems are learning to optimise profits rather than simply predicting monthly returns.

6.5 Summary and Conclusion

The application of machine-learning, genetic and evolutionary computing techniques to the real-world problem of portfolio optimisation is becoming increasingly realistic, and results from GP systems in particular are starting to show excellent profits. But is there anything intrinsically better about a GP system when compared with a popular machine-learning technique such as a Support Vector Machine (SVM)? In this paper a head-to-head comparison has been provided and has shown that the GP approach is qualitatively different to SVMs.

This investigation has included both a standard comparison of returns on investment and a comparison of both techniques when extended with a 'voting' mechanism designed to improve both returns and robustness to volatile markets. Robustness is an important additional dimension to this comparison, since the markets are dynamic and unpredictable. An investment simulator was used to model a long-short, market-neutral, sector-neutral hedge fund portfolio, trading Contracts for Difference (CFDs) in the highly volatile Malaysia stock market. Technical and fundamental historical stock data were used from the period 1997 to 2004.

It has been demonstrated that both a standard GP system and a voting GP system substantially outperforms its equivalent SVM on this problem. It is proposed that there is a key reason for this difference in performance — that the GP systems *learn to optimise profits* rather than simply predicting monthly stock returns. An experiment was conducted that demonstrated how GP systems provide solutions to this problem at a similar level of quality to SVMs when restricted to pure prediction of returns.

	Prediction	**Optimisation**
Voting	SVM-V: 0.22 pSGP-V: 0.04	SGP-V: 0.44
Non-Voting	SVM: -0.19 pSGP: -0.007	SGP: 0.11

Table 6.7: Sharpe ratios

These experiments have compared the performance of six different systems. Table 6.7 summarises the results by comparing Sharpe ratios. The dominance of the Optimisation/-Voting system (SGP-V) is clear, as is the general dominance of optimisation methods (SGP, SGP-V) over prediction methods (SVM, SVM-V, pSGP and pSGP-V). In conclusion, whilst the use of a voting mechanism is strongly beneficial to both SVM and GP systems in portfolio optimisation, GP's ability to integrate an overall portfolio profit optimisation with the evolution of a nonlinear stock ranking equation plays a vital role in generating profitable and robust solutions for use in volatile environments. This finding echoes perfectly the third unique advantage of GP highlighted in Section 2.1.2 that fitness functions add flexibility to the system by enabling the incorporation of problem-specific knowledge and encourage appropriate alternative behaviours.

Chapter 7

Conclusion

This chapter summarises the previous 6 chapters, reviews the contributions made, concludes with a discussion on the main results and finally suggest directions for future study in this area.

7.1 Summary of Thesis

This thesis has presented three novel GP algorithms designed to enhance the robustness of solutions evolved in highly dynamic environments and it has investigated the application of the new algorithms to financial time series analysis. The research is motivated by the following thesis question: what are viable strategies to enhance the robustness of GP solutions when the environment of a task being optimised or learned by a GP system is characterised by large, frequent and low-predictability changes?

The vast majority of existing techniques aimed to track dynamics of optima in very simple dynamic environments. But the research area in improving robustness in dynamic environments characterized by large, frequent and unpredictable changes is not yet widely explored. The three new algorithms were designed specifically to evolve robust solutions in these environments.

The first algorithm, 'behavioural diversity preservation' GP, is a novel diversity preservation technique for use in a Type 2 dynamic environment[1]. The algorithm evolves more robust solutions by preserving population phenotypic diversity through the reduction of behavioural intercorrelation and the promotion of individuals with unique behaviour.

The second algorithm, 'multiple-scenario training' GP, is a novel population training and evaluation technique for use in a Type 3 environment. The algorithm evolves more robust solutions by training a population across a set of pre-constructed environmental scenarios simultaneously and by using a 'consistency-adjusted' fitness measure to favour individuals performing well across the entire range of environmental scenarios.

The third algorithm, 'committee-voting' GP, is a novel 'final solution' selection technique for use in a Type 3 environment. The algorithm enhances robustness by creating a solution based on a majority-voting committee structure consisting of individuals evolved in a range of diverse environmental dynamics.

The thesis has introduced a comprehensive real-world case application for the evaluation

[1]See Section 3.1 for a discussion of different types of dynamic environments.

experiments. The case was a hedge fund stock selection application for a typical long-short market-neutral equity strategy in the Malaysian stock market. The underlying technology of the stock selection system was GP which assisted to select stocks by exploiting the underlying nonlinear relationship between diverse ranges of influencing factors. The three proposed algorithms were all applied to this case study during the evaluation experiments.

The results of experiments based on the case study demonstrated that solutions evolved by all three new algorithms are overwhelmingly more robust than those evolved by canonical GP.

The thesis has also presented the findings based an empiricial head-to-head comparison between GP and SVM for stock selection. The results demonstrated that GP substantially outperforms SVM and highlighted the advantage of using a fitness function in evolutionary approaches which allows for incorporating domain-specific requirements and knowledge.

7.2 Review of Contributions

This thesis has made the following contributions:

1. **The design and implementation of 3 novel GP algorithms for improving robustness of solutions**.

   (a) **'behavioural diversity preservation' GP**

   This algorithm was designed to address the problem of diversity preservation, more precisely phenotypic diversity preservation. The main design characteristics of the algorithm described in Section 3.3.1 are:

   - Preservation of diversity in a global population: Individuals in the population are classified based on their fitness values into hierarchical fitness segments whose size is constantly monitored during the evolution to prevent domination of a particular segment.
   - Preservation of diversity in fitness segments: An individual is deleted or preserved according to the degree of correlation of its phenotypic behaviour with other individuals in the same segment.
   - A random individual generator is used to continuously feed new genetic material into the lowest fitness segments.

   The full details of the algorithm implementation were given in Section 3.4.1. The details included:

   - Representation and calculation of fitness segments.
   - The mathematic formula of the correlation measure.
   - The rules on deletion and preservation of individuals in the fitness segments during evolution.
   - How a random individual generator was integrated in the evolution process.

(b) **'multiple-scenario training' GP**;

This algorithm was designed to address two of the problems identified - training and fitness evaluation. The main design characteristics of the algorithm described in Section 3.3.2 are:

- Parallel training of individuals in a set of different environmental scenarios.
- The selection of the individuals based on the mean-and-variance-adjusted ('consistency-adjusted') fitness evaluation which designed to favour individuals with good consistent overall performance across a set of different environments.

The full details of the algorithm implementation were given in Section 3.4.2. The details included:

- How a set of scenarios were selected and integrated in the GP evolution cycle.
- Mathematical formula used to represent 'consistency-adjusted' fitness.

(c) **'committee-voting' GP**.

Committee-voting GP was designed to address the problem of selecting a final solution. The main design characteristics of the algorithm are described in Section 3.3.3 are :

- The final solution consists of a set of individuals trained independently in a set of diverse environments.
- This set of individuals creates a decision-making committee for a given environment.

The full algorithm implementation details were given in Section 3.4.3. The details included:

- The two mechanisms for combining the solutions provided by each member in the committee -

2. **Validation of the novel algorithms by simulating a real-world hedge fund investment strategy with real-world data**.

The GP algorithms were validated by a real-world case study presented in Chapter 4. This case study focused on evolving multi-factor stock selection models for a long-short hedge fund invested in the Malaysia stock market. All the aspects of the case study were constructed in a real-world situation, including the data, factors, investment strategy and the portfolio trading process.

There were three main parts in this case study. The first was the factors. This work used 18 different factors, including fundamental, technical and size factors.

The second part was the investment simulator. The investment simulator was responsible for the execution of the hedge fund investment strategy. It traded on a monthly basis buying stocks to form the long portfolio and borrowing stocks to form the short portfolio for selling. Selection of long and short positions was based on a ranking measure which was generated by a GP system.

And the third part was the GP system. This was responsible for evolving factor models to generate quantitative attractiveness measures for stocks. The GP terminals were made of the factors and the functions were made of arithmetic functions plus several constants. A GP individual representing the underlying relationship between factors was expressed as a mathematic equation of the factors. The outputs of the system were the quantitative results derived from equations and they would be used to guide the investment simulator.

The investment simulator and the GP system were integrated together via the fitness function. The investment simulator was called each time the GP system needed to determine the fitness of an individual.

The validation involving measurement, comparison and analysis of the solutions evolved by the proposed GP algorithms and canonical GP, especially in terms of the three quantitative robustness measures, namely standard deviation, returns and rate of reduction of return (according to the new definition in Section 1.2) in the context of the case study, were summarised in Chapter 5.

3. **Formulating a new robustness defintion metrics**.

   A new definition of robustness developed by this work was formulated in Section 3.2 in the context of dynamic environments according to three criteria: (i) an overall measure of performance (fitness); (ii) a measure of the standard deviation of performances measured at each time interval (stability of fitness); (iii) the variation in overall performance between the training environment T and the unseen environment V (rate of fitness change).

   The new definition was used later in the thesis (Chapter 5 and Chapter 6) to quantitatively evaluate the robustness of solutions evolved by GP systems in each of these fundamental aspects.

4. **An empirical comparison and analysis of GP with a SVM**.

   The main objective of this 'early stage' empirical study in Chapter 6 was to shed some light on the qualitative differences between GP and other learning algorithms outside evolutionary approaches, such as SVM, in terms of the 'learning' mechanism in the context of the financial market. In this study, both the GP systems and SVM system were exposed to the Malaysian stock market during the long-short hedge fund simulation. In the first part the study, the GP systems used profit optimisation as the fitness function, and in the second part the GP systems used prediction accuracy (as used by SVM) as the fitness function. Two parameters were used to compare the systems: return on investment and robustness. The results clearly demonstrated that profit optimisation GP systems produced solutions which gave higher returns on investment and were more robust in dynamic environments than SVM systems. In contrast, the prediction GP systems did not offer advantages vis-*à*-vis the SVM system in either of the two parameters. The differing results obtained by the optimisation GP and prediction GP led to the conclusion that the qualitative difference between GP and SVM was GP's ability to integrate a (profit) optimisation mechanism into its learning mechanism.

7.3 Conclusion

In Chapter 1, the thesis null hypothesis was stated as

"When the environment of a task being optimised or learned by a GP system is characterised by large, sudden, frequent and low-predictability changes, the robustness of genetic individuals cannot be improved by means of the following algorithms.

Behavioural diversity preservation This preserves phenotypic diversity of a genetic population by reducing the degree of correlation between phenotypic behaviour of those individuals; greater phenotypic diversity leads to more robust solutions.

Multiple-scenario training This enhances robustness of solutions evolved by GP by means of training a population across a range of environment dynamics and selecting individuals that perform well in multiple scenarios;

Committee voting This enhances robustness of solutions evolved by GP by means of a committee structure whereby a small (odd) number of trained GP individuals offer solutions as votes, and the majority vote wins. "

In order to test the hypothesis, this thesis has implemented three new GP algorithms and has evaluated each in a real world application which simulates stock selection process for a long-short market-neutral hedge fund operating in the Malaysia stock market. Real historical stock data from 1997 to 2004 including price, technical factors and fundamental factors were used throughout the evaluation experiments. During this period, the Malaysia market underwent a long highly volatile phrase (Type 2 environment) and followed by a stabilising period (Type 3 environment). The results demonstrated that:

- In a Type 2 dynamic environment, it is viable to improve robustness of the solutions by preserving behavioural diversity of GP population.

  The results obtained provide an insight into the utility of the new algorithm (NGP) compared with a standard GP algorithm (SGP):

  1. on the test data, using robustness measure 2, NGP consistently produced a 'best' individual that was more robust with higher fitness than that produced by SGP without increasing volatility.
  2. on the test data, using robustness measure 3, NGP consistently produced a 'best' individual that was more robust with a lower rate of reduction of Sharpe ratio from training to out-of-sample test than that produced by SGP.

3. on the test data, in the context of continuous adaptation in a changing environment, NGP consistently produced a 'best' individual that was more robust than that produced by SGP using robustness measures 2 and 3.
4. on the test data, NGP consistently retrained (in the context of a shift in the environment) faster, and better, than SGP.

The results obtained also confirm that, on the test data, the new algorithm does what it intends — it increases the population diversity of phenotypic behaviour. It also increases diversity of standard fitness. Furthermore, analysis indicates that the difference from a standard GP algorithm is statistically highly significant.

It is concluded that (i) there is good evidence to recommend the use of NGP for preserving diversity of phenotypic behaviour *in any GP context*, and (ii) there is encouraging evidence to recommend the use of NGP for improving robustness in continuous adaptation contexts where the environment is continually changing.

- In an Type 3 environment, it is viable to obtain a substantial improvement of robustness of the solutions through 1) the use of carefully selected scenarios of market behaviour during GP training; and 2) the use of a voting committee comprising an odd number of GP individuals trained on a variety of different training sets (and therefore with differing phenotypic behaviour).

  For the multiple-scenario GP, experiments were run on three GP systems (MELG, MEVO and a 'standard' GP system — SGP) with 25 runs of each, and comparisons were made with both a portfolio index and a non-genetic simple technical analysis for stock picking.

  Although robustness measure 1 showed no significant difference between the three GP systems, statistical analysis of measures 2 and 3 indicated overwhelmingly that MELG provides the most robust individual, with SGP being the least robust. All three GP systems were shown to have better performance than the non-genetic technical analysis, and this in turn performed very much better than the portfolio index.

  For the committee voting GP, experiments were run on two GP systems: (i) SGP, and (ii) a committee of three 'best-of-run' individuals from three GP systems utilising different sets of training data. Statistical analysis indicated that the Voting system provides a remarkable improvement in terms of robustness measure 2 when compared to SGP.

  When the three best systems are compared — MELG, MEVO and the voting system — MELG and the voting system show a slight edge over MEVO in terms of all three robustness measures. The voting system appears to be slightly better than MELG; however, there is no statistically significant difference between MELG and the Voting system when the distributions of their Sharpe Ratios are compared.

In summary, the results produced in the experiments have rejected the null hypothesis and therefore, it is concluded that the proposed new algorithms are viable strategies for improving robustness of solutions evolved by GP in challenging dynamic environments.

7.4 Future Work

The literature survey in Chapter 2 observed that promoting robustness in EC techniques, especially for problems in dynamic environments, is a new area of research and that there are only a handful of works which implemented explicit robustness mechanisms. The new algorithms proposed in this thesis are, to date, the first to attempt to enhance the robustness of genetic individuals evolved by GP in extreme and complex real-world dynamic environments. Although the experiment results demonstrate that the new algorithms achieve the desired capabilities in Type 2 and Type 3 dynamic environments, the work is still in its early stages with many limitations, and there are many possible directions for future research.

7.4.1 Multiple-scenario training GP

One of the key features of the algorithm is the use of a fitness approximation function to calculate an individual's overall performance across a set of environmental scenarios. Fitness is a combination of standard deviation (representing consistency) and mean of a set of scenario fitnesses. For future research, the fitness approximation function can be extended to use other statistical methods to represent performance consistency.

Another direction is to investigate new ways to integrate a set of scenarios into the evolution process. This work proposed two ways: in one way the integration occurs at the beginning of the evolution process, generation 0, and in another the integration occurs at the last generation. The results showed that the second way was more effective. This may suggest that intensive training at a very early stage in the learning process is not necessary.

The algorithm used human intervention in selecting a set of environmental scenarios for training. One obvious extension is to fully automate this process.

7.4.2 Committee-voting GP

Committee member selection is one of the key aspects of the algorithm. This work used a 3-member committee and majority voting for the final decision. Future work in this area includes experimenting with different sizes of committees and different decision mechanisms, for example, integrating various assembly techniques developed in the area of machine statistical learning.

Another interesting direction is to investigate different ways to obtain good individual members with widely phenotypic behaviour, for example, the committee members may not necessarily need to be best-of-run individuals; or to investigate ways to optimise the combination of members, for example, the number of specialists (who excel in one type of environment) and the number of generalists (who perform consistently in all types of environments) required for an optimal combination.

One of the advantages of this algorithm is that it can be combined easily with other techniques. One possible extension is to combine the voting mechanism with the other two robustness techniques.

7.4.3 Behavioural diversity preservation GP

This algorithm uses Spearman correlation to measure the degree of correlation between an individual and its fitness segment. The obvious extension is to study and compare the impact of using different correlation measures. Future work could also undertake a parameter sensitivity analysis of the proposed system. The parameters are are the size of a fitness segment, number of random individuals generated at each generation and correlation threshold.

There are a number of existing diversity preservation works in the field, though they are not specifically designed for dynamic environments like this work. The evaluation of the algorithm can be extended to include comparison with some of the most well-known existing diversity preservation techniques.

7.4.4 The case study experiment

Because of great difficulty in collecting factor data mainly due to financial cost, the case study used 7-year data of 19 factors for 33 companies in one emerging market. Further work needs to extend the experiment to a larger universe of stocks over a longer period and to a different market to establish that these results are repeatable for a much wider range of test cases. Naturally, the experiment can also be extended to other appropriate areas of finance. One area of particular interest is to fully explore the capability of diversity algorithms in the context of online continuous-learning. This thesis was only able to initiate the discussion on this theme because of the lack of environment data.

7.4.5 Categorisation of dynamic environments

The categorisation of dynamic environments developed by this thesis was based on qualitative criteria. This left the categorisation quite broad. In order to develop more precise definitions, formulation of quantitative criteria for each environmental characteristic may be helpful. However, it would be extremely difficult to develop universal quantitative criteria because they are influenced by their domain characteristics, for instance, financial markets are very different from ecosystems etc. Furthermore, for some environments, many environmental criteria are difficult or impossible to quantify, the stock market is a such an example.

References

[1] Beware of bears. *The Economist* (online), October 15, 1998.

[2] Dancing in step. *The Economist* (online), May 22, 2001.

[3] Stockmarket swings. *The Economist* (online), Nov 17, 2001.

[4] Queasy feelings. *The Economist* (online), June 9, 2011.

[5] D. Achour, C. Harvey, and G. Hopkins. Stock selection in emerging markets: Portfolio strategies for malaysia, mexico, and south africa. *Emerging Markets Quarterly*, 2(4):38 – 91, 1998.

[6] V. Agarwal and N. Y. Naik. Risks and portfolio decision involving hedge funds. *Review of Financial Studies*, 17(1):63–98, 2004.

[7] E. Alfaro-Cid, K. Sharman, and A. Esparcia-Alcazar. A genetic programming approach for bankruptcy prediction using a highly unbalanced database. In *EvoWorkshops*, volume 4448 of *LNCS*, pages 169–178. Springer, 2007.

[8] P. Angeline. Adaptive and self-adaptive evolutionary computations. *Computational Intelligence: A Dynamic Systems Perspective, IEEE Press*, pages 152–163, 1995.

[9] K. M. Badran and P. Rockett. The role of diversity preservation and mutation in preventing population collapse in multiobjective genetic programming. *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 551–1558. ACM, 2007.

[10] S. Bandyopadhyay and S. K. Pal. *Classification and learning using genetic algorithms applications in bioinformatics and web intelligence (Natural Computing Series)*. Springer, ISBN 9783540496069, 2007.

[11] T. Bäck. Selective pressure in evolutionary algorithms: a characterization of selection mechanisms. *Proceedings of the First IEEE Conference IEEE World Congress on Computational Intelligence*, pages 57–62. IEEE Press, 1994.

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

[13] Y. Becker and U. M. O'Reilly. Genetic Programming for Quantitative Stock Selection. In *Procceedings of GEC 2009*, pages 9–15. ACM, 2009.

[14] K. Bennett and C. Campbell. Support vector machines: Hype or hallelujah. *SIGKDD Explorations*, 2(2): 200–208. ACM, 2003.

[15] T. F. Bersano-Begey and J. M. Daida. A discussion on generality and robustness and a framework for fitness set construction in genetic programming to promote robustness. In J. R. Koza, editor, *Late Breaking Papers at the 1997 Genetic Programming Conference*, pages 11–18, Stanford University, 1997.

[16] T. F. Bersano-Begey, J. M. Daida, J. F. Vesecky, and F. L. Ludwig. A Java collaborative interface for genetic programming applications: Image analysis and scientific inquiry. In *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, pages 477 – 482. IEEE Press, 1997.

[17] M. Brameier and W. Banzhaf. Explicit control of diversity and effective variation distance in linear genetic programming. In *Proceedings of the 5th European Conference on Genetic Programming*, volume 2278 of *LNCS*, pages 37–49. Springer, 2002.

[18] J. Branke. Memory enhanced evolutionary algorithms for changing optimisation problems. In *Proceedings of the 1999 Congress on Evolutionary Computations*, pages 1875–1882. IEEE, 1999.

[19] J. Branke. *Evolutionary Optimization in Dynamic Environments (Genetic Algorithms and Evolutionary Computation)*. Springer, 2001.

[20] J. Branke. *Optimisation in Dynamic Environments*. Kluwer Academic, 2002.

[21] J. Branke, T. Kauáler, and H. Scheck. *Guiding multi-objective evolutionary algorithms towards interesting regions*. Technical report. Institute AIFB, University of Karlsruhe, 2000.

[22] J. Branke, T. Kauáler, H. Scheck, and C. Schmidt. A multi-population approach to dynamic optimisation problems. In *Adaptive Computing in Design and Manufacturing*, pages 299–308. Springer, 2000.

[23] G. Brown, J. Wyatt, R. Harris, and X. Yao. Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5–20, 2005.

[24] G. Brown, J. Wyatt, R. Harris, and X. Yao. Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5–20, 2005.

[25] M. Butler and V. Keselj. Optimizing a pseudo financial factor model with support vector machines and genetic programming. In *Proceedings of Canadian AI 2009*, LNAI(5549): 191 – 194. Springer, 2009.

[26] R. Burke, S. Gustafson, and G. Kendall. A survey and analysis of diversity measures in genetic programming. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 716–723. Morgan Kaufmann, 2002.

[27] D. Z. Cao and S. Pang. Forecasting exchange rate using support vector machines. *Proceedsing of 2005 International Conference on Machine Learning and Cybernetics*, 6:3448 – 3452. IEEE, 2005.

[28] A. Carlise and G. Dozier. Adapting particle swarm optimisation to dynamic environments. *Proceedings of the International Conference on Artificial Intelligence* , pages 429–434, 2000.

[29] K. Y. Chen and C. H. Ho. An improved support vector regression modeling for Taiwan Stock Exchange market weighted index forecasting. In *International Conference on Neural Networks and Brain*, volume 3, pages 1633–1638. IEEE, 2005.

[30] S. H. Chen. On the relevance of genetic programming to evolutionary economics. In Aruka, editor, *Evolutionary Controvresy in Economics towards a New Method in Preference of Trans Discipline*, pages 135–150. Springer-Verlag, 2001.

[31] W. Chen and S. J. Comparison of support-vector machines and back propagation neural networks in forecasting the six major asian stock markets. *International Journal of Electronic Finance*, 1(1):49 – 67, 2006.

[32] H. Cobb. *An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments.* Technical report. Navy Centre for Applied Research in Artificial Intelligence, Naval Research Laboratory, Washington D. C., 1990.

[33] J. H. Cochrane. A cross-sectional test of an investment-based asset pricing model. *Journal of Political Economy*, 104(3):572, 1996.

[34] D. Costelloe and C. Ryan. On Improving Generalisation in Genetic Programming. *EuroGP 2009*, LNCS(5481):61–72, 2009.

[35] Commodity and Futures Trading Commission and Securities and Exchange Commission. *Findings regarding the market event of 6 may, 2010.* Report of the staffs of the CFTC and SEC to the Joint Advisory Committee on Emerging Regulatory Issues. September 2010.

[36] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press, 2000.

[37] P. Darwen and X. Yao. Why more choices cause less cooperation in iterated prisoner's dilemma. In *In Proceedings of the Congress on Evolutionary Computation*, pages 987–994. IEEE Press, 2003.

[38] D. Dasgupta. Using immunological principles in anomaly detection. In *Prceedings of the Artifical Neural Networks in Engineering*, pages 363–374, IEEE, 1996.

[39] E. D. de Jong, R. A. Watson, and J. B. Pollack. Reducing bloat and promoting diversity using multi-objective methods. In L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001)*, pages 11–18. Morgan Kaufmann, 2001.

[40] K. A. De Jong. Evolving in a changing world. In *International Symposium on foundations of intelligent systems*, pages 512–519. Springer, 1999.

[41] K. Deb and H. Gupta. Introducing robustness in multi-objective optimization. *Evolutionary Computation*, 14(4):463–494, 2006.

[42] M. A. H. Dempster and C. M. Jones. A real-time adaptive trading system using genetic programming. *Quantitative Finance*, 1:397–413, 2000.

[43] T. G. Dietterich. Ensemble methods in machine learning. In *First International Workshop on Multiple Classifer Systems*, pages 1–15. LNCS, Springer, 2000.

[44] T. G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, volume 1857 of *LNCS*, pages 1–15. Springer, 2000.

[45] A. E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transaction on Evolutionary Computation*, 3(2):124–141, 1999.

[46] A. Ekart and S. Nemeth. Maintaining the diversity of genetic programs. In *Proceedings of the 5th European Conference, Foster, J. et al., editors, Genetic Programming*, volume 2278 of *LNCS*, pages 162–171. Springer-Verlag, 2002.

[47] A. Ekart and S. Z. Nemeth. A metric for genetic programs and fitness sharing. In R. Poli, W. Banzhaf, W. B. Langdon, J. F. Miller, P. Nordin, and T. C. Fogarty, editors, *Genetic Programming, Proceedings of EuroGP'2000*, volume 1802 of *LNCS*, pages 259–270. Springer-Verlag, 2000.

[48] A. Esfahanipour and S. Mousavi. A genetic programming model to generate risk-adjusted technical trading rules in stock markets. *Expert Systems with Applications*, 38(2011): 8438-8445. Elsevier, 2011.

[49] B. Etzel. *Webster's New World Finance and Investment Dictionary.* Wiley Publishing, 2003.

[50] E. Fama. Efficient capital markets: A review of theory and emperical work. *Journal of Finance*, 25:383–417, 1970.

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

[52] A. Fan and M. Palaniswami. Stock selection using support vector machines. *Proceedings of IJCNN, International Joint Conference on Neural Networks*, 3:1793–1798. IEEE, 2001.

[53] T. C. Fogarty, F. Vavak, and P. Cheng. Use of genetic algorithm for load balancing for sugar beet presses. In *International Conference on Genetic Algorithms*, pages 617–624. Morgan Kaufmann, 1995.

[54] D. Fogel. The advantages of evolutionary computation. In *Proceedings of BCEC'1997*, pages 1–11. World Scientific Press, 1997.

[55] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423. Morgan Kaufmann, 1993.

[56] T. Friedrich, N. Hebbinghaus, and F. Neumann. Rigorous analysis of simple diversity mechanism. volume II of *GECCO 2007*, pages 1219–1225. ACM, 2007.

[57] A. Fukunaga. Restart scheduling for genetic algorithms. *Proceedings of the 7th International Conference on Genetic Algorithms*, pages 357–358, IEEE, 2008.

[58] C. Gagné, M. Schoenauer, M. Parizeau, and M. Tomassini. Genetic programming, validation sets, and parsimony pressure. In P. Collet, M. Tomassini, M. Ebner, S. Gustafson, and A. Ekárt, editors, *Proceedings of the 9th European Conference on Genetic Programming*, volume 3905 of *Lecture Notes in Computer Science*, pages 109–120. Springer, 2006.

[59] C. Gathercole and P. Ross. An adverse interaction between crossover and restricted tree depth in genetic programming. Genetic Programming 1996, First annual conference, pages 291–296. MIT press, 1996.

[60] A. Gierer, S. Berking, H. Bode, C. N. David, K. Flick, G. Hansmann, H. Schaller, and E. Trenkner. Regeneration of hydra from reaggregated cells. *Nature New Biology*, 239:98–101, 1972.

[61] D. Goldberg. *Genetic Algorithms in Search, Optimisation, and Machine Learning.* Addison-Wesley, 1989.

[62] D. Goldberg and R. Smith. Nonstationary function optimisation using genetic algorithms with dominance and diploidy. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 59–68. Lawrence Erlbaum Associates Inc. 1987.

[63] J. Grefenstette. Genetic algorithms for changing environments. In *Parallel Problem Solving from Nature*, Vol 2, pages 137–144, 1992.

[64] A. M. Gupal and A. M. Tsvetkov. Committee-based inductive inference. *Cybernetics and Systems Analysis*, 28(5):785–787. Springer, 1992.

[65] S. Gustafson. *An analysis of diversity in genetic programming.* PhD thesis, University of Nottingham, 2004.

[66] M. Hashem Pesaran and A. Timmermann. A recursive modelling approach to predicting uk stock returns. *The Economic Journal*, 110(Jan):159–191. Blackwell Publishers, 2000.

[67] T. Haynes, R. Wainwright, S. Sen, and D. Schoenefeld. Strongly typed genetic programming in evolving cooperation strategies. In L. Eshelman, editor, *Genetic Algorithms: Proceedings of the Sixth International Conference (ICGA95)*, pages 271–278. Morgan Kaufmann, 1995.

[68] J. L. Hippolyte, C. Bloch, P. Chatonnay, C. Espanet, and D. Chamagne. A self-adaptive multiagent evolutionary algorithm for electrical machine design. *Proceeding of GECCO 2008*, pages 1251–1255. ACM, 2007.

[69] C. Hommes. Economic agents and markets as emergent phenomena : Modeling the stylized facts in finance through simple nonlinear adaptive systems. *PNAS*, 99:7221–7228. Springer, 2002

[70] N. N. Horn, J. and D. E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Computation*, pages 82–87. IEEE, 1994.

[71] G. S. Hornby. Alps – the age-layered structure for reducing the problem of premature convergence. *Proceedings of the 8th annual conference on genetic and evolutionary computation*, 1:815 – 822. ACM, 2006.

[72] J. How, M. Ling and P. Verhoeven. Does size matter? A genetic programming approach to technical trading. *Quantitative Finance*, 10(2):131 – 140. Taylor & Francis, 2010.

[73] J. Hu. *Sustainable Evolutionary Algorithms and Scalable Evolutionary Synthesis of Dynamic Systems.* PhD thesis, Michigan State University, 2004.

[74] X. Hu and R. Eberhart. Adaptive particle swarm optimisation: Detection and response to dynamic systems. In *Proceedings of the 2002 Congress on Evolutionary Computation*, pages 1666–1670. IEEE, 2002.

[75] W. Huang, Y. Natamori, and S. Y. Wang. Forecasting stock market movement direction with support vector machines. *Comupter and Operation Research*, 32(10):2513 – 2522, 2005.

[76] M. Hutter. Self-optimizing and Pareto-optimal policies in general environments based on Bayes-mixtures. In *Proc. 15th Annual Conf. on Computational Learning Theory (COLT'02)*, volume 2375 of *LNAI*, pages 364–379. Springer, 2002.

[77] K. Imamura, T. Soule, R. Heckendorn, and J. Foster. Behavioral diversity and a probabilistically optimal GP ensemble. *Genetic Programming and Evolvable Machines*, 4:235–253, 2003.

[78] T. Ito, H. Iba, and M. Kimura. Robustness of robot programs generated by Genetic Programming. In J. R. Koza, D. E. Goldberg, D. B. Fogel, and R. L. Riolo, editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, Stanford University, pages 321–326. MIT Press, 1996

[79] B. Jacobs and K. Levy. *Equity Management – Quantitative Analysis for Stock Selection.* McGraw-Hill, 2000

[80] E. Jordaan, A. Kordon, L. Chiang, and G. Smits. Robust inferential sensors based on ensemble of predictors generated by genetic programming. *PPSN VIII, LNCS 3242*, pages 522–531. Springer, 2004.

[81] C. L. Karr. Adaptive process control suing biologic paradigms. In *Proceedings of Electronic Technology Directions to the Year 2000*, pages 128 –136. Elsevier, 1995.

[82] V. Kecman. *Learning and Soft Computing – Support Vector Machines, Neural Networks and Fuzzy Logic Models.* MIT Press, 2001.

[83] M. Keijzer. Improving Symbolic Regression with Interval Arithmetic and Linear Scaling. In *Proceedings of EuroGP 2003*, LNCS (2610): 70–82. Springer, Heidelberg, 2003.

[84] R. E. Keller and W. Banzhaf. Genetic programming using genotype-phenotype mapping from linear genomes into linear phenotypes. In *Genetic Programming 1996: Proceedings of the First Annual Conference*, Stanford University, pages 116 – 122. MIT Press, 1996.

[85] J. Kennedy and R. Eberhart. Particle swarm optimisation. Proceedings of IEEE Int'l Conf. on Neural Networks, pages 1942–1948. IEEE, 1996.

[86] J. Kim and P. Bentley. The artificial immune model for network intrusion detection. In *7th on European Congress Intelligent Techniques and Soft Computing(EUFIT'99)*, IEEE, 1999.

[87] J. Kim and P. Bentley. Towards an artificial immune system for network intrusion detection: An investigation. In *Proceedings of the 2001 Congress on Evolutionary Computation*, Vol 2, pages 1244 –1252. IEEE, 2002.

[88] K. E. Kinnear and Jr. Generality and difficulty in genetic programming: Evolving a sort. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 287–294. Morgan Kaufmann Publishers , 1993.

[89] H. Kitamo. *Foundations of systems biology.* MIT Press, ISBN 0-262-11266-3, 2001.

[90] H. Kitano. Cancer robustness: tumour tactics. In *Nature*, 426(6963):125. Nature Publishing Group, 2003.

[91] H. Kitano. Biological robustness. In *Reviews*, 5(11):826–837, Nature Publishing Group, 2004.

[92] V. Koumousis and C. Katsaras. A saw-tooth genetic algorithm combining the effects of variable population size and reinitialisation to enhance performance. *IEEE Transaction on Evolutionary Computation*, 10(1):19–28, 2006.

[93] J. Koza. *Genetic programming: on the programming of computers by means of natural selection.* MIT Press, 1992.

[94] J. Koza. *Genetic Programming II Automatic Discovery of Reusable Programs.* MIT Press, ISBN 0-262-11189-6, 1994.

[95] J. Koza. *Genetic Programming: Routine Human-Competitive Machine Intelligence (Genetic Programming).* Kluwer Academic Publishers, june edition, 2003.

[96] J. Koza, M. Keane, and M. J. Streeter. *Genetic Programming IV: Routine Human-Competitive Machine Intelligence.* Springer, 2005.

[97] I. Kushchu. Generalisation and domain specific functions in genetic programming. In *Proceedings of the 2000 Congress on Evolutionary Computation, CEC 2000*, volume 2, pages 1393–1400. IEEE Press,2000. .

[98] I. Kushchu. Promoting generalisation of learned behaviours in genetic programming. PPSN V, LNCS 1498, pages 491–500. Springer-Verlag Berlin Heidelberg, 1998.

[99] I. Kushchu. An evaluation of evolutionary generalisation in genetic programming. *Artificial Intelligence Review*, 18(1):3–14, Sept. 2002.

[100] I. Kushchu. Genetic programming and evolutionary generalization. *IEEE Transactions on Evolutionary Computation*, 6(5):431–442, Oct. 2002.

[101] H. Lam. T. Bui, D. Essam and D. G. Green. Performance analysis of evolution multi-objective optimisation algorithms in noisy environments. *Complexity International*, 11:29–39, 2005.

[102] W. Langdon and R. Poli. *Foundations of Genetic Programming.* Springer-Verlag, 2002.

[103] W. B. Langdon. *Genetic Programming and Data Structure: Genetic programming + data structure = automatic programming!*. Kluwer Academic Publisher, 1998.

[104] W. B. Langdon and R. Poli. An analysis of the max problem in genetic programming. In Proceedings of Genetic Progamming 1997, the second International Conference on Genetic Programming, pages 222–230. Morgan Kaufman, 1997.

[105] B. LeBaron, W. B. Arthur, and R. Palmer. The time series properties of an artificial stock market. *Journal of Economic Dynamics and Control*, 23:1487–1516, 1999

[106] S. Legg and M. Hutter. Fitness uniform deletion: A simple way to preserve diversity. In *Proc. Genetic and Evolutionary Computation Conference, GECCO 2005*, pages 1271–1278. ACM, 2005.

[107] J. Li and E. Tsang. Improving technical analysis predictions: An application of genetic programming. In *Proceedings of Florida Artifical Intelligence Research Symsosium, FLAIRS 99*, pages 108–112. IEEE, 1999.

[108] Y. Liu, X. Yao, and T. Higuchi. Evolutionary ensembles with negative correlation learning. *IEEE Transactions on Evolutionary Computation*, 4(4):380, 2000.

[109] J. Lohn, G. Hornby, and D. Linden. Human-competitive Evolved Antennas. *AIEDAM: Artificial Intelligence for Engineering, Design, and Manufacturing*, 22:235–247, 2008.

[110] D. Lohpetch and D. Corne. Discovering Effective Technical Trading Rules with Genetic Programming: Towards Robustly Outperforming Buy-and-Hold. *NaBIC: 2009 World Congress on Nature and Biologically Inspired Computing*, 439–444, 2009.

[111] R. Lung and D. Dumitrescu. A new evolutionary model for detecting multiple optima. In *Proceedings of GECCO 2007*, pages 1296–1303. ACM, 2007.

[112] S. Mahfoud and G. Mani. Financial forecasting using genetic algorithms. *Applied Artificial Intelligence*, vol 10(6): 543-565, 1996.

[113] S. Mahfound. Boltzmann selection. In T. Bӓck, D. Fogel, and Z. Michalewicz, editors, *Handbook of evolutionary computation*. Oxford University Press, New York, 1997.

[114] H. M. Markowitz. Portfolio selection:efficient diversification of investment. *Mita Business Review*, 3(1):120–125, 1959.

[115] H. M. Markowitz and A. F. Perold. Portfolio analysis with factors and scenarios. *The Journal of Finance*, 36(4):871–877.

[116] R. McKay. Fitness sharing in genetic programming. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2000*, pages 435–442. ACM, 2000.

[117] R. Mckay and H. Abbass. Anticorrelation measures in genetic programming. In *Australian-Japan Workshop on Intelligent and Evolutionary System*, pages 45–51. Springer, 2001.

[118] J. F. Miller. Evolving a self-repairing, self-regulating, french flag organism. In *Proceedings of Genetic and Evolutionary Computation Conference, GECCO 2004*, pages 129–139. ACM, 2004.

[119] T. M. Mitchell. *Machine Learning.* McGraw-Hill, 1997.

[120] F. W. Moore and O. N. Gacia. A new methodology for reducing brittleness in genetic programming. In *Proceedings of the National Aerospace and Electronics 1997 Conferences, NAECON-97*, pages 757–763. IEEE, 1997.

[121] R. W. Morrison. *Designing Evolutionary Algorithms for Dynamic Environments.* Natural Computing Series. Springer, 2004.

[122] C. Neely, P. Weller, and R. Dittmar. Is technical analysis in the foreign exchange market profitable? A genetic programming approach. *Journal of Financial and Quantitative Analysis*, 32(4):404–426, 1997.

[123] V. Nissen and J. Propach. On the robustness of population-based versus point-based optimisation in the presence of noise. *IEEE Transactions on Evolutionary Computation*, 2(3): 107–119, 1998.

[124] V. Nissen and J. Propach. Optimization with noisy function. In *Proceedings of Parallel Problem Solving in Nature (PPSN V)*, pages 159–168. Springer, 1998.

[125] F. Norris. It May Be Outrageous, but Wall Street Pay Didnt Cause This Crisis. *The New York Times*, 31 July 2009.

[126] M. O'Neill, L. Vanneschi, S. Gustafson, and W. Banzhaf. Open issues in genetic programming. *Genetic Programming and Evolvable Machines*, 11(3):339–363, Sept. 2010.

[127] U. M. O'Reilly. *An analysis of genetic programming.* PhD thesis, Carelton University, Ottawa, Ontario, Canada, 1995.

[128] W. T. Pan. The use of genetic programming for the construction of a financial management model in an enterprise. *Appl. Intell.*, 36(2): 271-279, 2012.

[129] L. Panait and S. Luke. Methods for evolving robust programs. In *Proceeding of Genetic and Evolutionary Computation Conference, GECCO 2003*, volume 2724 of *LNCS*, pages 1740–1751. Springer, 2003.

[130] D. Parrott and X. Li. A particle swarm model for tracking multiple peaks in a dynamic environment using speciation. In *Proceedings of the Congress on Evolutionary Computation, CEC 2004* , vol 1, pages 98–103. IEEE, 2004.

[131] A. Piszcz and T. Soule. Dynamics of evolutionary robustness. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation, GECCO 2006*, pages 871–878. ACM, 2006.

[132] W. Langdon and R. Poli. *A field guide to genetic programming.* (paperback). Lulu, 2008.

[133] C. Reeves and H. Karata. Dynamic sequencing of a multi-processor system: a genetic algorithm approach. In *Artificial Neural Nets and Genetic Algorithms*, pages 491–495. Springer, 1993.

[134] C. W. Reynolds. An evolved, vision-based behavioral model of coordinated group motion. In Meyer and Wilson, editors, *Proceedings of Second Conference on From Animals to Animats 2: Simulation of Adaptive Behaviour.* pages 384–392. MIT Press, 1992.

[135] P. Rohlfshagen, P. Lehre, and X. Yao. Dynamic evolutionary optimisation: An analysis of frequency and magnitude of change. *Proceedings of the 2009 Genetic and Evolutionary Computation Conference (GECCO 2009)*, 2009.

[136] J. Rosca. Entropy-driven adaptive representation. In *Proceedings of Workshop Genetic Programming: From Theory to Real-World Applications, J. Rosca, Ed.*, pages 23–32. Morgan Kaufmann, 1995.

[137] J. Rosca. Genetic programming exploratory power and the discovery of function. In *Evolutionary Programming IV Proceedings of the Fourth Annual Conference on Evolutionary Programming*, pages 719–736. MIT Press, 1995.

[138] C. Ryan. Pygmies and civil servants. In K. Kinnear, Jr., editor, *Advances in Genetic Programming*, chapter 11, pages 243–263. MIT Press, 1994.

[139] S. Salcedo-Sanz, J. L. Fernandez-Villacanas, M. J. Segovia-Vargas, and C. Bousono-Calzon. Genetic programming for the prediction of insolvency in non-life insurance companies. *Computers & Operations Research*, 32(4):749–765, 2005.

[140] B. Scholkopf and A. Smola. *Learning with Kernels – Support Vector Machines Regulerization, Optimization and Beyond.* MIT Press, 2002.

[141] W. Sharpe. Risk, market sensitivity, and diversification. *Financial Analysts Journal (January/February 1972)*, pages 74–79, 1972.

[142] W. F. Sharpe. Capital asset price: a theory of market equilibrium under condition of risk. *Journal of Finance*, 19(3):425–442, 1964.

[143] W. F. Sharpe. The sharpe ratio. *Journal Portfolio Management*, 21:49–58, (Fall) 1994.

[144] C. Silva and B. Ribeiro. *Inductive Inference for Large Scale Text Classfication - kernel approaches and techniques.* Springer–Verlag Berlin, 2010.

[145] B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, 1986.

[146] A. Simões and E. Costa. Using genetic algorithm to deal with dynamic environments: A comparative study of several approaches based on promoting diversity. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2002*, pages 697–704. Morgan Kaufmann, 2002.

[147] J. Smith and T. Fogarty. Operator and parameter adaptation in genetic algorithms. *Soft Computing*, 1(2):81–87, 1997.

[148] R. Smith. Adaptively resizing populations: an algorithm and analysis. In *Proceeding of the 5th International Conference Genetic Algorithms*, page 653 – 660. Morgan Kaufmann Publishers, 1993.

[149] D. Sornette. *Why Stock Markets Crash: Critical Events in Complex Financial Systems.* Princeton University Press, 2002.

[150] D. Sornette and A. Johansen. Large financial crashes. *Physica A – statistical Mechanics and Its Applications*, 245(3–4):411–422, 1997.

[151] T. Soule. Voting teams: A cooperative approach to non-typical problems using genetic programming. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 99*, volume 1, pages 916–922. Morgan Kaufmann, 1999.

[152] T. Soule. Cooperative evolution on the intertwined spirals problem. In C. Ryan, T. Soule, M. Keijzer, E. Tsang, R. Poli, and E. Costa, editors, *Genetic Programming, Proceedings of EuroGP 2003*, volume 2610 of *LNCS*, pages 434–442. Springer-Verlag, 2003. .

[153] T. Soule, R. B. Heckendorn, and J. Shen. Solution stability in evolutionary computation. In *Proceedings of ISCIS XVII Seventeenth International Symposium on Computer and Information Services*, pages 237–241. ACM, 2002.

[154] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

[155] H. Subramanian, S. Ramamoorthy, P. Stone, and B. Kuipers. Designing safe, profitable automated stock trading agents using evolutionary algorithms. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation, GECCO 2006*, pages 1777–1784. ACM, 2006.

[156] A. Surette, M. G. Barkai, and S. Leibler. Robustness in bacterial chemotaxis. In *Nature*, volume 397, pages 168 – 171. Nature Publishing Group, (January) 1999.

[157] N. N. Taleb. *The Black Swans.* Penguin Books, 2007.

[158] K. Tan, T. Lee, and E. Khor. Evolutionary algorithms with dyanmic population size and local exploration for multiobjective optimisation. *IEEE Transactions on Evolutionary Computation*, 5(6):565–588, 2001.

[159] K. Trojanowski and Z. Michalewicz. Evolutionary algorithm for non-stationary environments. In *Proceedings of 8th Workshop: Intelligent Information systems*, pages 229–240. ICS PAS Press, 1999.

[160] G. Trujillo, Leonardo; Olague. Synthesis of interest point detectors through genetic programming. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 887-894. ACM, 2006

[161] A. Turing. Can a Machine Think. in Newman, James edition *The World of Mathematics*, volume 4, pages 2099 – 2133. Simon & Schuster, 1956.

[162] A. Turing. Intelligent Machinery. *Cybernetics: Key Papers*, eds. C. R. Evans and A. D. J. Robertson, University Park Press, Baltimore Md. and Manchester, 1968.

[163] R. Ursem. Multinational gas: Multimodal optimisation techniques in dynamic environment. In *In Proceedings of the Second Genetic and Evolutionary Computation Conference, GECOO 2000*, pages 19–26. Morgan Kaufmann, 2000.

[164] R. K. Ursem. Diversity-guided evolutionary algorithms. In J. Guervos, P. Adamidis, H. Beyer, J. Martin, and H. Schwefel, editors, *Proceedings of the 7th international Conference on Parallel Problem Solving From Nature*, volume 2439 of *LNCS*, pages 462–474. Springer-Verlag, 2002.

[165] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Varlag, 1999.

[166] F. Vavak and K. Jukes. Performance of a genetic algorithm with variable local search range relative to frequency of the environmental changes. In *Genetic Programming 1998: Proceedings of the Third Annual Conference* , pages 602–609. Morgan Kaufmann, 1998.

[167] A. S. Vieira, B. Ribeiro, S. Mukkamala, J. C. Neves, and A. H. Sung. On the performance of learning machines for bankruptcy detection. In *Proc. 2nd IEEE International Conference on Computational Cybernetics, ICCC'04.*, pages 323–327. IEEE, 2004.

[168] A. Wagner. *Robustness and Evolvability in Living Systems.* Princeton University Press, 2005.

[169] P. Wang, E. Tsang, T. Weise, K. Tang and X. Yao. Using GP to Evolve Decision Rules for Classification in Financial Data Sets. In *Proc. 9th IEEE Int. Conf. on Cognitive Informatics (ICCI10).*, pages 720–727. IEEE, 2010.

[170] H. M. Wibbur and R. Volker. Life-history evolution in uncertain environments: Bet hedging in time. *The American Naturalist*, 168(3):398–411, 2006.

[171] R. Wiegand and M. Potter. Robustness in cooperative coevolution. In volume 1 of *Proceedings of the 8th annual conference on Genetic and evolutionary computation, GECCO 2006*, pages 369–376. ACM, 2006.

[172] M. Wineberg and J. Chen. The shifting balance genetic algorithm as more than just another island model GA. In *Proceedings of the 6th annual conference on Genetic and evolutionary computation, GECCO 2004*, pages 318–329. Springer-Verlag Heidelberg, 2004.

[173] D. H. Wolpert and W. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[174] W. Yan and C. D. Clack. Diverse committees vote for dependable profits. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation, GECCO 2007*, pages 2226–2233. ACM, 2007.

[175] W. Yan and C. D. Clack. Behavioural GP diversity for dynamic environments: an application in hedge fund investment. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation, GECCO 2007*, pages 1817–1824. ACM, 2006.

[176] W. Yan and C. D. Clack. Diverse committees vote for dependable profits. In *Proceedings of 9th annual conference on Genetic and evolutionary computation*, volume 2, pages 2226–2233. ACM, 2007.

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

[178] W. Yan and C. D. Clack. Learning to optimize profits beats predicting returns - comparing techniques for financial portfolio optimisation. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation, GECCO 2008*, pages 1681–1688. ACM, 2008.

[179] W. Yan and C. D. Clack. Behavioural GP diversity for adaptive stock selection. In *Proceedings of the 11th annual conference on Genetic and evolutionary computation, GECCO 2009*, 2009.

[180] W. Yan and C. D. Clack. Evolving robust GP solutions for hedge fund stock selection in emerging markets. *Soft Computing – Special issue on Bio-inspired Learning and Intelligent Systems*, 15(1), pages 37–50. Springer Berlin, 2010.

[181] S. Yang. Memory-based immigrants for genetic algorithms in dynamic environments. volume 3 of *Proceedings of the 7th annual conference on Genetic and evolutionary computation GECCO 2005*, pages 1115–1122, Springer, 2005.

[182] S. Yang. A comparative study of immune system based genetic algorithms in dynamic environments. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation, GECCO 2006*, pages 1377–1384. ACM, 2006.

[183] S. Yang and X. Yao. Experimental study on population-based incremental learning algorithms for dynamic optimisation problems. *Soft Computing*, 9(11):815–834, 2005.

[184] B. T. Zhang and J. G. Joung. Building optimal committees of genetic programs. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature – PPSN VI*, pages 231–240, Berlin, 2000. Springer.

[185] D. Zhang, H. Huang, Q. Chen, and Y. Jiang. A comparison study of credit scoring models. In *Proc. Third International Conference on Natural Computation (ICNC 2007)*, pages 15–18. IEEE, 2007.

[186] E. Zitzler, M. Laumanns, L. Thiele, and C. M. Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.

[187] E. Zitzler, M. Laumanns, L. Thiele, C. M. Fonseca, and V. G. da Fonseca. Why quality assessment of multiobjective optimizers is difficult. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 02*, pages 666–674. Morgan Kaufmann, 2002.

[188] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271. IEEE, 1999.