Testing Market Imperfections via Genetic Programming

Dissertation zur Erlangung des Doktorgrades
der Wirtschaftswissenschaften (Dr. oec.)

Fakultät Wirtschafts-
und Sozialwissenschaften
Universität Hohenheim

Lehrstuhl für Bankwirtschaft
und Finanzdienstleistungen

vorgelegt von
Sebastian Jansen

aus Bonn

2010

Dekan:	Prof. Dr. Dirk Hachmeister
1. berichtende Person:	Prof. Dr. Hans-Peter Burghof
2. berichtende Person:	Prof. Dr. Stefan Kirn
3. berichtende Person:	Prof. Dr. Günter Schmidt
Eingereicht am:	03.05.2010
Mündliche Prüfung am:	17.03.2011

Die vorliegende Arbeit wurde am 11. Februar 2011 von der Fakultät Wirtschafts- und Sozialwissenschaften der Universität Hohenheim als Dissertation zur Erlangung des Doktorgrades der Wirtschaftswissenschaften angenommen.

Danksagung

Zu Beginn der Dissertation möchte ich die Möglichkeit nutzen, mich ausdrücklich bei denjenigen zu bedanken, die entscheidend zur Entstehung der vorliegenden Arbeit beigetragen haben. Zu allererst gilt mein Dank Prof. Dr. Hans-Peter Burghof, der sich bereit erklärt hat, die Arbeit zu betreuen. Seiner fachlichen Neugier und Aufgeschlossenheit ist es zu verdanken, dass überhaupt ein scheinbar ausgetretenes Sujet wie Markteffizienz noch einmal Gegenstand einer Dissertation geworden ist, in diesem Fall mit einer zumindest erklärungsbedürftigen Methodik namens Genetic Programming, mit der man üblicherweise eher Dinge des technischen Alltags wie Ampelschaltungen oder Tragflächen von Flugzeugen optimiert. Prof. Burghof hat meine anfangs noch recht unklare Projektskizze vom Testen evolutionärer computergenerierter Handelssysteme in die finanzwirtschaftliche Forschung (in diesem Fall Testen von Markteffizienz) überführt, mich motiviert, immer wieder Denkanstöße gegeben und die Arbeit aus der ein oder anderen Sackgasse geführt. Weiterhin möchte ich mich herzlich bei Prof. Dr. Stefan Kirn bedanken, der sich bereit erklärt hat, als Zweitgutachter aufzutreten. Mein Dank gilt ebenfalls Prof. Dr.-Ing. Günter Schmidt vom Lehrstuhl für Informations- und Technologiemanagement der Universität des Saarlandes für wertvolles Feedback zu meiner Arbeit und für sein Drittgutachten. Aus technischer Sicht gebührt Sara Silva von der Evolutionary and Complex Systems Group der University of Coimbra in Portugal mein Dank. Ohne ihre Software GPLAB wäre diese Arbeit vermutlich unmöglich gewesen. Besonders hervorzuheben ist auch ihre sehr schnelle Reaktion auf einen Bug, der das Projekt zu einem vorübergehenden Stillstand verdammt hatte. Außerdem danke ich Herrn Jan Christoph Kinne für seine zahlreichen klugen Denkanstöße aus Sicht eines Mathematikers, die mir geholfen haben, mein eigenes Projekt besser zu verstehen und zu hinterfragen. Last but not least möchte ich mich herzlich bei meinen Eltern bedanken, die mich stets moralisch (und finanziell) während meiner gesamten Ausbildung kompromisslos unterstützt haben.

Bonn, April 2010

List of Tables

List of Figures

List of Abbreviations

AA/TS	Asset Allocation/Trading Systems
ANN	Artifical Neural Network
AR	Auto-Regressive
ARCH	Auto-Regressive Conditional Heteroscedasticity
ARMA	Auto-Regressive Moving Average
CAPM	Capital Asset Pricing Model
CPU	Central Processing Unit
DD	Downside Deviation
EA	Evolutionary Algorithms
EC	Evolutionary Computing
EMH	Efficient Markets Hypothesis
ETF	Exchange Traded Funds
EURIBOR	Euro Interbank Offered Rate
FIBOR	Frankfurt Interbank Offered Rate
FOREX	Foreign Exchange
FS	Fuzzy Systems
GA	Genetic Algorithms
GA/GP	Genetic Algorithms/Genetic Programming
GARCH	Generalized Auto-Regressive Conditional Heteroscedasticity
GP	Genetic Programming
GPLAB	Genetic Programming Laboratory
HIBOR	Hong Kong Interbank Offered Rate
KBES	Knowledge-Based Expert System
LISP	List Processor
MA	Moving Average
MAR	Minimum Acceptable Rate of Return
MTS	Mechanical Trading System
TTR	Technical Trading Rule

Contents

Part I

Introduction + Motivation

1 Evolutionary Algorithms

Evolutionary Algorithms (EA) are tools for heuristic optimization based on simulation of evolutionary processes in Nature. While the class of EA comprises several subclasses of algorithms which will be briefly addressed later, so-called Genetic Algorithms (GA) and Genetic Programming (GP) have emerged as the two most widely used techniques. After the pioneering theoretical framework was introduced by Holland (1975), GA gradually made their way from theoretical biology to applied mathematics, physics, chemistry, computer science and engineering. Based on Holland's foundation, Koza (1992) introduced GP in order to refine the evolutionary approach to optimization problems. Applications of GA and GP are manifold, ranging from applications as diverse as minimization of sonic boom on supersonic aircraft (Karr et al., 2003) and traffic signal timing optimization (Sun et al., 2003) to evolutionary optimization of molecular docking (Yang, 2003), a component of rational drug design, to name just a few. A glimpse of the wealth of real-world applications is presented in Cantú-Paz (2003).

During the last couple of years, GA and GP have become an important tool in economics and finance as well. GA and GP constitute a promising approach to modeling the highly complex dynamics of financial markets and numerous articles on GA and GP with applications to finance have been published. However, the total number of publications is surprisingly low compared to other topics in finance such as artificial neural networks, behavioral finance or credit risk. Although the reasons may be numerous, it is quite likely that, especially in terms of evolutionary modeling of trading strategies, considerable efforts are made at private institutions such as banks. Given the assumption that GA/GP are a suitable tool for modeling and forecasting financial asset returns, approaches that prove to be profitable in some way remain, for obvious reasons, undisclosed.

This might partly explain the somewhat sporadic and fragmented research in this field. Nevertheless, a sufficient amount of material has been published and several articles have found their way into prestigious journals thus underlining the suitability and acceptance of the approach among the academic community.

The thesis is organized as follows: The next chapter presents a literature review followed by a thorough discussion of the basics of GP in the third chapter. The fourth chapter presents the setup and results for the application of GP-optimized trading rules to the DAX and the Hang Seng. The final chapter provides a summary and conclusion.

2 Motivation

First of all, it must be emphasized that this section is just a very brief introduction and that the points made here are elaborated upon later in the thesis. Basically, the motivation for the thesis at hand is, unsurprisingly, to test whether stock markets are efficient. The efficient markets hypothesis (EMH) was first formulated by Fama (1970) and roughly speaking says that the participants in a financial market efficiently use all information so that all information is priced into the market in such a way that no profits from a particular trading strategy should be in excess of a passive buy-and-hold investment in the same market[1]. While the EMH was widely accepted in academics at first, a considerable number of papers have questioned the validity of the EMH. In their seminal paper, Brock et al. (1992) reported profitable trading strategies for the S&P 500 which sparked further research into the validity of the EMH. Profitable trading strategies for stock markets have also been reported by Jegadeesh and Titman (1993), Bessembinder and Chan (1995), Huang (1995) and Kwon and Kish (2002), to name a few. LeBaron (1999) also reported successful trading rules in the FOREX market as did Raj and Thurston (1996) for futures. These findings are seriously shaking the assumption of efficient markets, even in its weakest form (assumed impossibility to forecast returns based on past prices of

[1] This definition is rather imprecise. An exact definition of the EMH will be given in the fourth chapter.

the underlying security)[2]. From a GP point of view, the paper by Brock et al. (1992) and the follow-up literature were at the core of an entirely new branch in the EMH literature that is dedicated to finding trading rules by means of evolutionary optimization. The basic line of reasoning was that if it was possible to find trading rules with econometric techniques it might also be worthwhile to do so using methods from computional intelligence such as the already existing GA introduced by Holland (1975) and the new technique GP introduced by Koza (1992). The advent of GP emphasized the capabilities of optimization techniques inspired by evolutionary processes found in Nature. Furthermore, computers had become powerful enough to deal with computationally demanding applications such as GP. The first attempts at using GA/GP were made by Bauer (1992, 1994), Neely et al. (1997) and most notably Allen and Karjalainen (1999). However GA/GP-related financial market research is still quite limited as will be seen in the upcoming literature review. Particularly striking is the lack of thorough research in terms of particular stock markets since most of the existing contributions focus on U.S. markets. Therefore, the thesis at hand extends the literature to other major indices such as the German DAX and Hong Kong's Hang Seng and checks whether GP can provide an answer to one of the major topics in finance, i.e. are markets efficient?

[2] Admittedly, there are also opposing points of view concerning profitability of trading strategies such as Hudson et al. (1996), Bessembinder and Chan (1998), Brown et al. (1998) and most notably Chen and Kuo (2001).

Part II

Applications of Evolutionary Algorithms in Asset Allocation and Trading Systems

3 Introduction

This chapter aims at reviewing the current state of literature on applications of GA/GP in finance, with emphasis on asset allocation and trading systems. Hybrid models, i.e. crossing EA with competing techniques such as neural networks for example, will be considered as well. Last but not least, the existing literature on GA/GP based forecasting will also be covered as it is closely linked to the search for profitable trading systems using GA/GP. The remainder of the chapter is organized as follows: As a precursor to a discussion of applications of GA/GP in asset allocation and trading systems (AA/TS), several computer-aided trading systems will be presented in the fourth section. At this stage, the mechanics of GA/GP will be outlined as well, albeit in a very brief fashion[3]. The fifth section constitutes the mainstay of this chapter and reviews the literature on GA/GP-applications in AA/TS. The section ends with a brief discussion of hybrid models and applications of GA/GP in forecasting.

4 Computer-Based Trading Systems

The aim of the upcoming section is to introduce the concept of technical trading rules as building blocks for computer-based trading systems (with one of them being GA/GP) which will then be briefly discussed.

[3] A full-scale presentation of GA/GP including a thorough discussion of all parameters involved are the main topic of the third chapter of the thesis.

4.1 Some Remarks on Technical Trading Rules

Technical trading rules (TTR) are a set of rules used by traders and portfolio managers to buy and sell securities on financial markets. Basically, the idea is to predict future prices based on past prices. The probably most well-known rule is the moving average rule as shown in equation (13.1) which comes in various flavours such as the 200-day moving average advocated by Granville (1976). Apart from this classic indicator, other inputs such as daily high/lows, trading volume and volatility are also used frequently. A compilation of some popular rules is found in Babcock (1989). TTR are widely used by market practicioners. In contrast to this, in academia financial markets were believed to follow a random walk (Fama, 1965a, 1965b) thus rendering any trading rule useless from a theoretical point of view. The closely related EMH questioned any gains from particular trading patterns as well (Fama, 1970). Although several publications reported profitable trading strategies such as Basu (1977) and French (1980) for stock markets and Sweeney (1986) for the foreign exchange market (FOREX), it was not until 1988 that Lo and McKinlay showed that markets do not follow random walks, a fact that was taken for granted until then by practitioners. As TTR can be implemented by computer systems, several computer-based trading systems exist, with one of them being GA/GP. Examples of an algorithm capable of detecting classic trading patterns such as head-shoulder-head are presented in Lo et al. (2000). An overview of the general requirements for a suitable trading system and the design process is given in Pardo (1992).

4.2 Computer-Aided Trading Systems

Sophisticated and thus computer-based techniques such as GA/GP are available to develop trading models. The upcoming discussion aims at giving an overview of these techniques. However, a thorough discussion of the strengths and weaknesses of each technique is beyond the scope of the thesis. An account of computer-based trading systems is given in Figure 4.1.

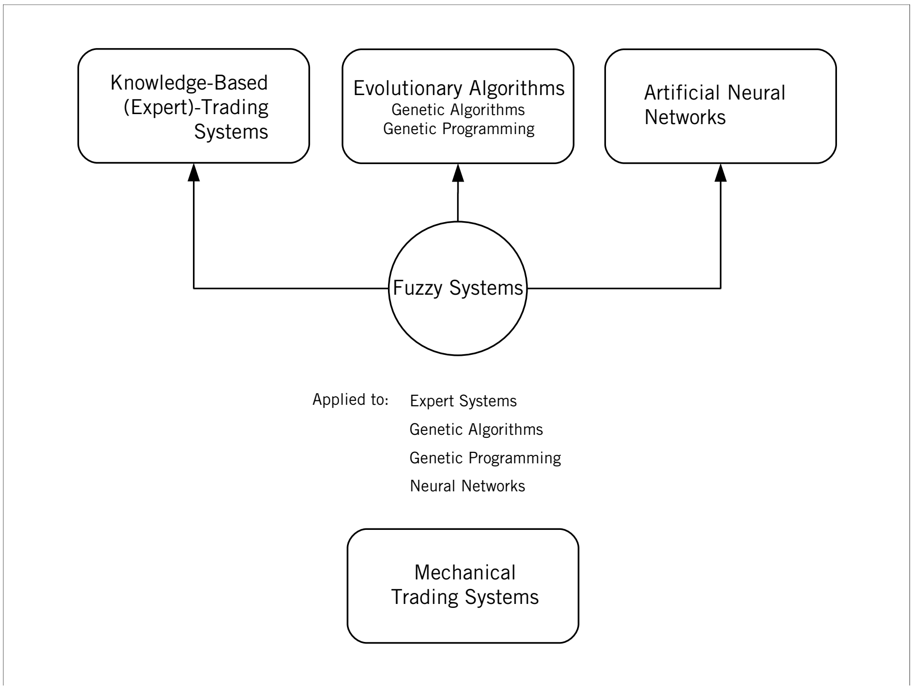


Figure 4.1: *An overview of computer-based trading systems.*

4.2.1 Knowledge-Based Expert-Trading Systems

With reference to Medsker (1995), Knowledge-Based Expert-(Trading) Systems (KBES) perform reasoning using a set of previously established rules. These rules are stored in a knowledge base and are fed-forward to a so-called inference engine which then provides the end-user with advice. As an example, Deboeck (1994) reports that a large brokerage firm set up a KBES by collecting not less than 600 trading rules from their traders. The system was then employed to provide inference based on these rules to assist in trading operations. The case illustrates the main drawback of KBES, i. e. high complexity and difficult maintenance since the knowledge-base has to be checked for consistency/validity and has to be updated frequently.

4.2.2 Mechanical Trading Systems

Mechnical Trading Systems (MTS) are a more complex implementation of TTR. An arbitrarily large array of rules can be easily implemented with the help of a computer system. The system then uses simple if-then-reasoning to generate buy- and sell-signals which can either be passed on to the trader (if the system is designed as an advice-giving support platform) or can be executed directly via computer. Deboeck (1994) presents a combination of 5- and 20-day moving averages for the S&P 500 as an example of a very basic mechanical trading system. Varying levels of complexity can be used to improve trading performance. However, as Deboeck (1994) points out, MTS are generally prone to overfitting. He reports that the majority of trading systems are not very profitable from a historical point of view, at least in terms of profitability vs. risk.

4.2.3 Artificial Neural Networks

Artifical Neural Networks (ANN) try to imitate biological neural networks as those found in human brains. ANN are made up of three main components: The inputs x, followed by hidden layers n and an output y. A typical so-called feed-forward network is depicted in Figure 4.2. The neurons n can be thought of as electrical impulses that are triggered by the inputs x. The neurons then fire an impulse which results in an output y. The neuron firing mechanism is triggered by an activation function. The weighted sum of inputs serves as input to this activation function. The three most common activation functions are the so-called sigmoid, tansig and Gaussian activation function as depicted in Figure 4.3. The exact functional forms can be found in McNelis (2005). Each activation function has the common attribute of triggering a response once a certain threshold value has been exceeded just in analogy to biological systems. A single snowflake on a bare hand does not trigger the feeling of cold whereas many snowflakes in sum trigger this sensation. If the threshold value is not exceeded, the activation function literally remains silent. ANN are a quite established technique. They are particularly interesting because it can be shown that they are capable of approximating any nonlinear function to infinitely accurate precision (McNelis, 2005). Although ANN differ substantially from GA/GP, they are in a certain sense a direct competitor to the latter as they serve the same purpose, i.e. modeling nonlinear dynamics. Although the question which approach is more suitable for modeling financial markets is intriguing it would deserve a thesis in its own right. Therefore, this issue will not be addressed further. However, the best of two worlds can be combined

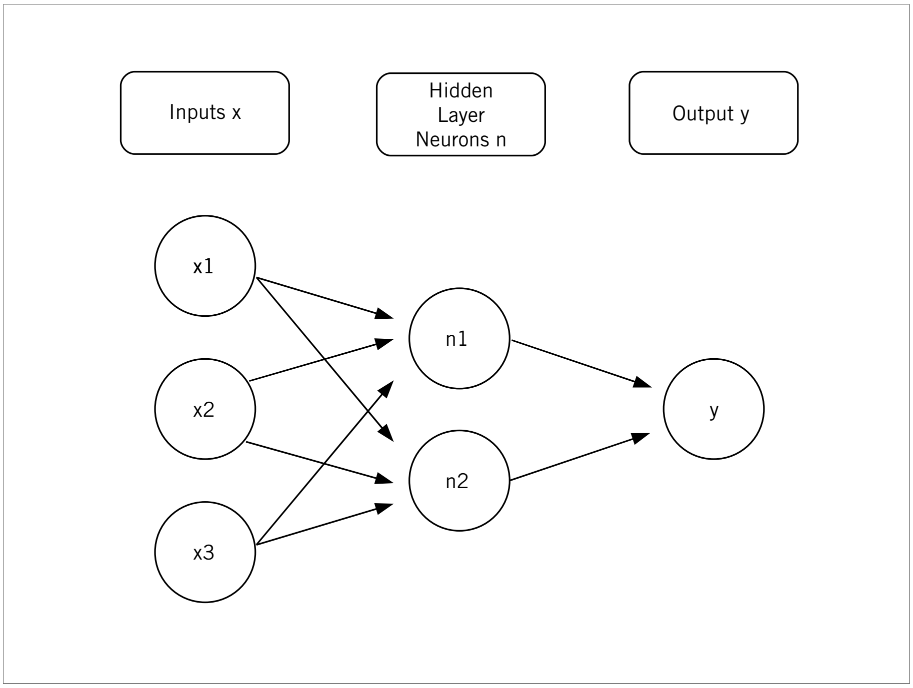


Figure 4.2: *A typical one-layer feed-forward neural network.*

in a promising way as will be shown later when GA/GP-optimized ANN also known as Neuro-Genetic hybrid models will be presented. It is safe to say that the literature on ANN is abundant. A glimpse of the sheer amount of papers available is given in McNelis (2005).

4.2.4 Fuzzy Trading Systems

As indicated in Figure 4.1, Fuzzy Systems (FS) are some sort of support vehicle capable of enhancing the power of KBES, ANN and GA/GP. They were pioneered by Zadeh (1965) and are based on so-called fuzzy set theory. Broadly speaking, FS try to emulate some kind of approximate reasoning. Input to these models is given in fuzzy, i. e. imprecise terms. The output can be either fuzzy or precise. FS therefore are capable of emulating human decision making which is often qualitative rather than quantitative. As an illustration, Medsker (1995) picks “hot” and “cold” as fuzzy inputs. What exactly is meant by “hot” and “cold” can be described by a so-called membership function which, for example, could state that “hot” means everything in between 30 °C-40 °C. The system

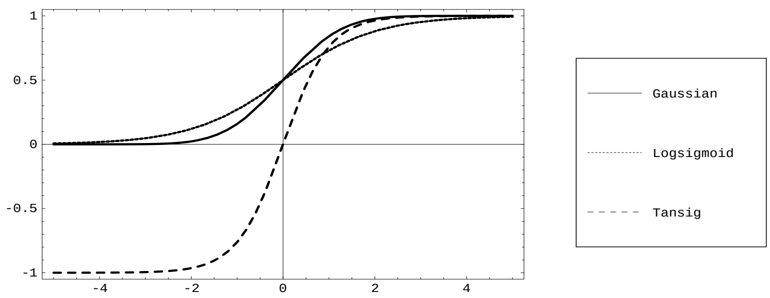


Figure 4.3: *Three common activation functions for Artificial Neural Networks.*

can be used vice versa as well by entering exact temperatures as input in order to get the reasoning "hot" or "cold" as output. In analogy to that, a simple FS for trading could be based upon inputs such as "good" or "bad" stock market performance. Based on this reasoning, the system could then generate a simple buy or sell signal. As said before, FS mainly serve as a support tool for KBES, ANN and GA/GP rather than a stand-alone system which is why they will not be elaborated upon further. However, two hybrid Fuzzy-GA papers will be briefly presented later.

4.2.5 Evolutionary Algorithms

The class of EA will be outlined in the third chapter of the thesis so for now, suffice it to say that GA and GP have probably become the most widely used techniques within evolutionary computing.

4.2.5.1 Genetic Algorithms

Having explained all other computer-based trading systems, it is now time to introduce the so-called GA for the first time. GA are search algorithms that emulate evolutionary processes in Nature. They were first introduced by Holland (1975) and belong to the class of heuristic optimization techniques. Based on the Darwinian survival-of-the-fittest theme, GA attempt to find an optimal solution to a problem by starting with a randomly generated set of potential solution candidates which are encoded in a binary string consisting of "0"s and "1"s. The solutions are then evaluated and ranked according to their individual fitness. The most promising solutions are selected and merged to form the next generation. During this process, random mutations occur to ensure that the search process covers a wide set of the search space. To put it in a nutshell, a basic GA works as follows:

1. Random generation of potential solutions to a problem.
2. Calculation of their respective fitness.
3. Select best solutions, merge (crossover) and apply mutation.
4. Re-evaluate fitness , select best candidates.
5. Iterative repetition of (3) and (4) until no further improvements in fitness can be achieved.

As Goldberg (1993) points out, GA are particularly appealing as an optimization technique since, unlike analytical approaches, they do not impose any requirements such as continuity and existence of derivatives on the underlying function to be optimized.

4.2.5.2 Genetic Programming

An important extension of the Holland (1975) GA is so-called GP first introduced by Koza (1992). Basically, GP incorporates the main attributes of GA, i.e. efficient search of the solution space by applying a fitness measure (excess returns for example) to the solution candidates that are subject to operators like crossover and mutation. An important difference is that GP solutions, unlike GA, are not represented by binary strings of fixed length but via tree-like structures for each solution. A simple example of such a solution tree (which has to be checked for fitness) is depicted in Figure 4.4 using technical indicators as input. Trees are read from bottom to top. The simple tree reads: "Buy the stock/index if the average stock price over the past 50 days is greater than the current price p or the current transaction volume v is less than 20".

A new and perhaps fitter solution tree can be generated by, for example, discarding the right volume-related subtree in Figure 4.4 and replacing it with an arbitrary subtree of a similar complex tree that the GP setup randomly creates during initialization (crossover). The new "child-tree" can then be evaluated

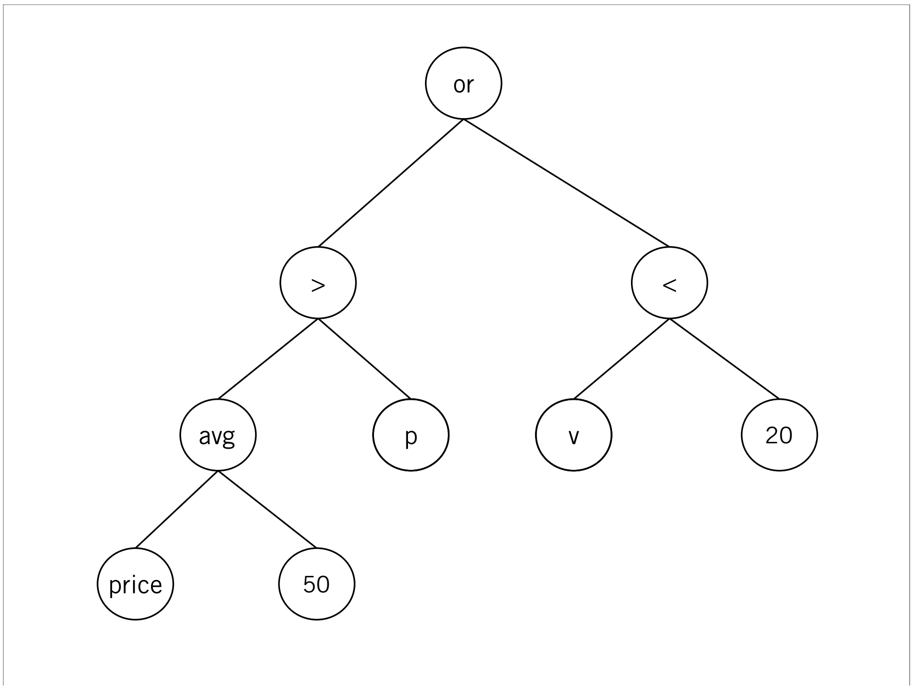


Figure 4.4: *A technical trading rule in Genetic Programming tree-like structure taken from Potvin et al. (2004).*

once more. The tree in Figure 4.4 is just based on a limited range of operators. An overview of operators available to GP will be given in the third chapter of the thesis. Solution trees can vary in complexity (that is in depth of subtrees and number of nodes) and are generally more flexible than GA especially when the structure of the solution is not known a priori. For example, GA can only operate with a fixed amount of variables whereas a GP approach can vary the amount of variables and indicators used allowing for a more flexible design.

Now that all computer-based trading systems as shown in Figure 4.1 have been briefly reviewed, it is time to discuss applications of GA/GP in finance.

5 Genetic Algorithms/Programming in Finance

This section constitutes the main part of the chapter. An overview of applications of GA/GP in finance is given in Figure 5.1. The author decided to merge AA/TS into a single category since it is somewhat difficult to draw the line be-

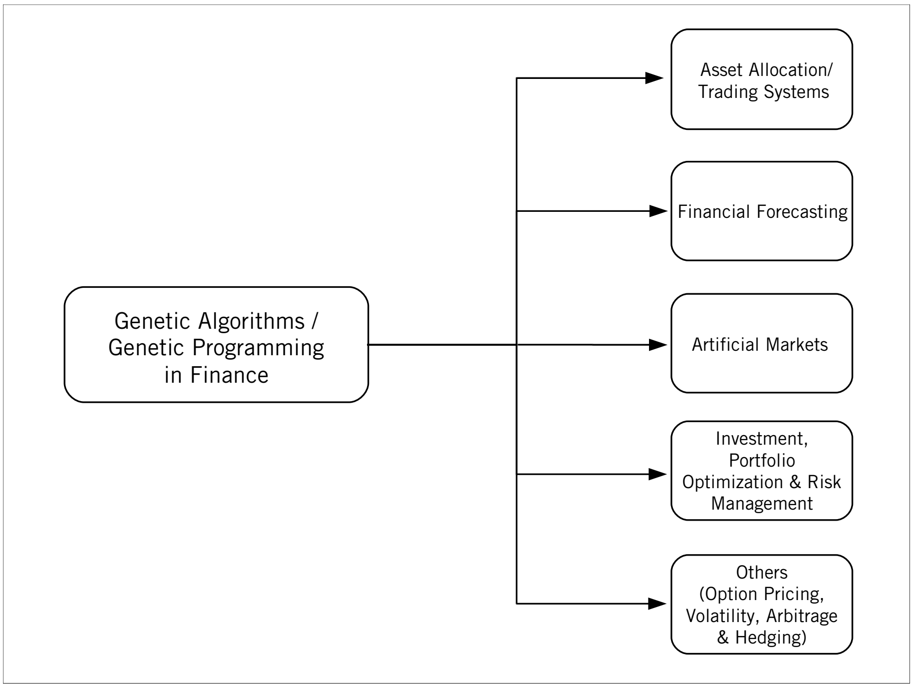


Figure 5.1: *Applications of Genetic Algorithms and Genetic Programming in finance.*

tween the two fields. A GA/GP-based trading system, i.e. a set of trading rules that provides traders with buy- and sell-signals can be used by fund managers for (tactical) asset allocation as well. The same works vice-versa: A GA/GP-powered tactical asset allocation scheme can be used for fund management and trading floor operations alike. The only difference might be that trading systems are designed to execute numerous trades a day or even high-frequency trading whereas tactical asset allocation, despite the designation "tactical", rather refers to mid-term strategies with less frequent trading compared to the fast-paced action on the trading floor. Later on in this chapter, the subsection on forecasting is meant as a supplement as either fields (AA/TS and forecasting) are to some extent intertwined. It is obvious that a GA/GP-based forecasting approach can be exploited to set up a trading system. The same applies to GA/GP-trading systems as well. A system providing the end-user with buy- and sell-signals is to some degree a forecasting system as well[4]. Returning to Figure 5.1, the

[4]However, as Yu, Chen and Kuo (2004) point out, a profiable trading system can be a poor forecasting system since it might only be profitable by (randomly) picking up large moves in the underlying market while being on the wrong side of the market most of the time.

lower three fields will not be elaborated upon further. They are covered in Chen (2002a, 2002b).

5.1 Genetic Algorithms/Programming in Asset Allocation and Trading Systems

Although GA were devoloped in the seventies of the last century and further developed in the eighties, it was not until the nineties that they found their way into AA/TS. The most likely reason for this might be the assumption of financial markets following random walks and the EMH which, on a theoretical level, contradicted profits from trading rules. This paradigm was seriously questioned by Lo and McKinlay (1988) who showed that markets do not follow random walks and by Brock et al. (1992) with their seminal article. Brock et al. (1992) tested popular trading rules over a 90-year horizon in the S&P 500. The rules included 20 different moving average rules and six versions of the trading-range break rule. They found that both classes of rules work well which translates into buy signals generating 12% annual return on average and sell signals generating 7% loss per year on average. These findings seriously contradicted the EMH. Brock et al. (1992) laid the groundwork for further research into trading models and eventually GA/GP for AA/TS. Since then, several contributions made their way into prestigious journals thus underlining the suitability and acceptance of the approach among the research community.

5.1.1 Stock Markets

The first attempt (to the best of the author's knowledge) at creating a tactical asset allocation scheme was made by Bauer and Liepins (1992). They illustrate the usefulness of GA by providing a fund switching example. They assume that an investor can either invest 100% of his assets in an S&P 500 fund or alternatively in a small-firm equity fund on a monthly basis. The investment horizon is five years and the investor aims to maximize terminal wealth. An investment strategy can be translated into a binary string consisting of "0"s (invest in S&P 500) and "1"s (invest in small-firms fund) for every month within a five year horizon. Therefore, $2^{12x5} \approx 1$ quintilion strategies exist. Since the performance of either investment is ex-post known, the optimum strategy can be easily calculated by hand by simply determining for every month within the 5-year horizon which investment performed better. The data used ranged from 1926-1985 and was split into consecutive 5-year subsamples. The authors find that their GA quickly converged to a (near-) optimum in all twelve 5-year subperiods. The optimal solution was found in 7 out of 12 periods, for the five remaining subperiods the solution yielded by the GA was in excess of 99% of the optimum on average. The authors then modify the algorithm to account for transaction costs which is easily implemented. Although the optimal solution was now found in only 43 out of 120 trials compared to 105 out of 120 trials in the first example, GA still proved powerful. In the 77 remaining cases, the solution was always within a 10% margin of the optimum. As a further refinement, the authors add two additional investment alternatives to the investor's set of choices, namely long-term government bonds and Treasury bills, making computation more cumbersome. Nevertheless, GA was still able to find near optimal solutions which were 94.5% on average of the optimal solution. Although the study can be critizised on grounds of being too simplistic, lack of out-of-sample testing and most notably ex-post data snooping, it still shows the power and flexibility of GA as a tool for tactical asset allocation.

Bauer (1994) presents a comprehensive account of GA-driven investment strategies. He focuses on stock and bond markets[5]. One of the most striking features of the study is, like Ammann and Zenkner (2003), the use of macroeconomic variables as input to the GA whereas most of the literature relies on technical indicators such as (lagged) prices, moving averages etc. as will be seen later. Bauer (1994) picks ten macroeconomic variables found to have the highest correlation with excess returns in the S&P 500 over the T-bill; among them indices for U.S. inflation, production levels and unemployment. The benchmark is a classic buy-and-hold strategy in the S&P 500, the alternative investment is a long position in (virtually) default-free T-bills. The training data ranges from 1984-1988. The resulting trading rules are applied out-of-sample from 1989-1992. Typical trading rules look like: If inflation > (<) *threshold value* AND (OR) production level < (>) *threshold value* OR (AND) unemployment > (<)

[5]The results for the U.S. government and U.S. corporate bond market will be covered later.

threshold value, then buy the index, else invest in T-bills[6]. For the 1989-1992 holdout period, the author reports a negative excess return on average (14.96% vs. 18.46% for buy-and-hold). But he also finds that, though unable to beat buy-and-hold, the trading rules significantly reduced risk as the rules were on average only 8 months long in the markets on a yearly basis instead of being exposed over the whole 12 months like buy-and-hold. As a result, the portfolio made up of GA-generated trading rules earned 81% of the buy-and-hold benchmark but with only 75% percent of the associated risk. Furthermore, a hedge portfolio consisting of a portfolio with the best trading rules and a portfolio with the worst trading rules was constructed. The idea was to go long on the good rules and to go short on the bad rules. The author finds that the long portfolio outperformed the short portfolio for the entire holdout period opening up avenues for profitable investments. In conclusion, although the Bauer (1994) GA did not beat buy-and-hold in the S&P 500, it reduced risk by a significant amount.

An interesting variation is Frick et al. (1996) who, as an intriguing feature, use Frankfurt stock exchange data (1989-1994) for their study. Another feature setting their paper apart from others is that inputs to the GA are based on a popular heuristic method called point & figure charts. Basically, this type of chart depicts the presence and strength of price reversals for a particular stock or entire index[7]. The setup in the study first creates appropriate point & figure charts based on historic price data of each share in the DAX which are then converted into a binary representation. By combining the data extracted from the individual charts, resistance and support levels for each share can be computed and trading rules can be created. The performance of these rules is then compared with the riskless rate/market return and the expected, risk-adjusted return within the established CAPM framework for the time frame considered. If the return was higher than the just mentioned benchmarks, a buy-signal was emitted, otherwise a sell signal. The DAX served as a proxy for market return and the FIBOR[8] was used as the riskless rate. The authors report an average

[6] With the recent advent of Exchange Traded Funds (ETF), it is possible to buy an entire index directly. Therefore, the problem translates into when to buy an ETF.

[7] The exact procedure is described in Tölke (1992).

[8] Frankfurt Interbank Offered Rate.

winning percentage of 60%, i.e. the buy- and sell-signals based on GA-powered trading rules were correct 60% of time on average with single rules being correct in excess of 70% of time which illustrates the potential power of GA. However, performance is found to degrade over time during the out-of-sample test period. Unfortunately, the authors do not investigate the profitability of their findings and do not elaborate further on the results of their study.

Kassicieh et al. (1997) adapt the Bauer (1994) approach to find optimal switching strategies between the S&P500 and T-bills on a monthly basis using macroeconomic inputs with the highest correlation to the S&P 500. Based on the data sample (1958-1993), the authors find that GA performance in terms of terminal wealth is close to that of the ex-post known perfect switching strategy between the two asset classes.

Fyfe et al. (1999) focus on a single stock, namely a property investment firm called *Land Securities plc* to check whether profitable GP-trading rules exist. The data range from 1980-1997, technical indicators were used as input to GP. The GP approach succeeds in finding a profitable trading rule that beats the buy-and-hold benchmark. Overall profit during the entire holdout period was 407.8% vs. 335.5% for buy-and-hold. Further analysis showed that the most profitable rule had never issued a sell signal (although the authors report that this almost had been the case during the october 1987 crash) and instead only took long positions for certain periods. Therefore, the authors term the rule "timing-specific buy-and-hold" referring to the fact that the rule found is nothing more than a slightly more sophisticated buy-and-hold rule. Based on these findings, the authors conclude that the market (at least the market for *Land Securities plc*) is quite efficient[9].

Allen and Karjalainen (1999) use GP to develop a trading system for the S&P 500. The data set covers 1929-1995. The algorithm was designed to find optimal trading rules on a daily basis and yields in-the-market and out-of-the-market

[9] It might be considered a weakness of the study that it just focuses on a single listed stock rather than a wider selection of stocks or an entire index. This shortcoming was addressed in Fyfe et al. (2005).

signals which translates into "buy-the-index" and "stay-out-of-the-market and earn the risk-free rate". The rules are compared with a standard buy-and-hold strategy. Technical indicators were used as input such as moving averages and trading range breaks. The setup allowed for a free search of parameters in the solution space. Artifical indicators such as a 183-day moving average could emerge during GP runs. Therefore, it was up to the GP to find out the optimal length of a moving average or exact numerical specification of a trading range break resulting in more flexible trading rules. To guard against data snooping, a 5-year training period was selected followed by a 2-year validation period during which the best rules accumulated thus far were tested again. The final selection was then applied out-of-sample to the rest of the data until 1995. With realistic transaction costs, the algorithm was unable to consistently outperform the benchmark. However, the authors show that the timing strategies have some forecasting ability as volatility is lower when the strategies indicate to be in-the-market compared to out-of-the-market days. Averaged over all trading rules and out-of-sample periods, the volatility of annual trading rule returns is 10% opposed to 14.1% for the S&P 500 during the same period. Furthermore, the authors report that volatility can be further reduced by setting up a portfolio of rules to diversify risk. If an equal amount of capital is put in each of the strategies found by GP for a particular trading period, volatility can be further reduced to 8.7% on average. As a consequence, even though the rules fail to beat the market, the authors argue that the notably lower volatility might appeal to investors on a risk-adjusted basis[10]. Due to the lack of consistent outperformance of the timing strategies vis-à-vis buy-and-hold, the authors conclude that the EMH holds.

Bhattacharyya and Mehta (2002) develop a GP-trading system for the S&P 500 as well. High, low, closing prices, moving averages, and variances of high and low prices for succeeding time windows were chosen as input for the algorithm. The data ranged from 1983-1997. The authors report an average excess return over the buy-and-hold-benchmark of 4.41% for the out-of-sample period after ten years of training. Interestingly and consistent with Bauer (1994), Allen and

[10] The volatility-reducing effect will be subject to further investigation as part of this thesis.

Karjalainen (1999) and Ammann and Zenkner (2003), the power of the timing strategies is reported to diminish during prolonged out-of-sample application indicating major structural breaks in the underlying market dynamics[11].

Pereira (2002) looks at the Australian stock exchange ASX to test a GA framework. The data ranged from 1982-1997, technical indicators were taken as inputs to the GA. Typical transaction costs of 10 basis points were considered. On a risk-adjusted basis, the trading rules found almost consistently outperform the buy-and-hold benchmark during the out-of-sample test period. However, the profitability of the trading rules is found to decline over time. In addition, a refinement to the methodology to account for thinly traded shares (so-called non-synchronous trading/return measurement bias) lead to a meltdown of risk-adjusted excess returns. As an interesting result and in-line with Allen and Karjalainen (1999), the author notes that the trading rules are long in the market when volatility of returns is low whereas they tend to stay out of the market when volatility is high, indicating some timing/forecasting potential of the rules.

An innovative approach to GP-trading is presented in Thomas and Sycara (2002). The design of their GP setup allows for the inclusion of stock-specific messages posted on internet message boards. The message volume on two boards, namely *YAHOO!* and *Lycos Finance* is taken as input to the algorithm. The data are based on the top 10% by internet message traffice volume of the Russell 1000 index ranging from Jan. 1998 until Dec. 2001 (68 stocks in total). As a first step, the message traffic data for each share was collected resulting in a new time series of message traffic for each stock. The GP setup was tasked to yield buy- or sell-signals based on a pre-defined threshold level of message data. The idea was that once a certain threshold level had been exceeded, a rare (negative) stock-specific event had occured which should be interpreted as a sell signal. If a single stock is shorted, a long position in a broader market, i. e. the Russell 1000 is taken. The benchmark to the switching strategy between individual stocks and a broad index was a simple buy-and-hold strategy in the appropriate stock. The results of the study are positive: While the buy-and-hold

[11] The main focus of the paper is on the choice and impact of different fitness functions and lengths of training and out-of-sample periods on GP-performance.

strategy earned 126.21% over the entire test period, the GP approach earned 164.36%. The Sharpe ratio is reportedly superior as well (1.15 vs. 1.74915). Using a bootstrap test, the authors show that their results are statistically significant. As part of further anaylsis, the authors checked whether the internet message traffic just echoed information contained in other data as well such as lagged trading volume or lagged return. They find, despite some correlation between the variables, that internet message traffic does contain unique information about the underlying stock. They conclude that the inclusion of "soft" factors such as message board traffic seems promising as part of a GP-based trading system.

Becker and Seshadri (2003a) pick up the setup and results from Allen and Karjalainen (1999) and fine-tune their search algorithm in different ways. They use monthly rather than daily data to reduce trading frequency, different fitness measures and most importantly reduce the complexity of the search space by restricting the amount of operators and indicators used for GP. The training period ranges from 1960-1990 and the resulting rules are tested from 1991-2002. The benchmark investment was once more a long position in the index. Interestingly, the authors find that the leaner and improved algorithm succeeds in consistently outperforming the buy-and-hold benchmark in the out-of-sample period at a statistically signficant level. Unfortunately, their report is rather brief and therefore they do not elaborate further on their results. As a conclusion, it seems that (overly) complex GP implementations result in sub-optimal performance.

Another study is Ammann and Zenkner (2003). Based on five macroeconomic variables, namely interest rate spreads, default spreads, dividend returns, GNP and inflation for the U.S., the authors try to find an optimal asset allocation scheme. Assets can either be invested 100% in the S&P 500 or 100% in 3-month T-Bills which are virtually risk-free. As a benchmark, a standard buy-and-hold strategy was chosen. Based on data ranging from 1980-2000, the strategy to be derived should point out on a daily basis whether to invest in the market or not. The ratio of in-sample years to out-of-sample years was 5:5, 5:1 and 5:10, i.e. five years of training data applied to the next 5 years out-of-sample and so on. The GA yielded an excess return of 3.47% during the eighties accompanied by a Sharpe ratio of 1.17 vs. 0.66 for the buy-and-hold benchmark. In contrast to this, the GA performs worse during the nineties and yields slightly negative excess returns. The authors explain this finding by referring to different market conditions and structural breaks. While the GA performs well during the volatile eighties, the sustained long-term upward trend during the nineties seems to favour the buy-and-hold strategy which, by definition, is always long in-the-market. However, the timing strategy derived by the GA yields slightly better Sharpe ratios (0.71 vs. 0.68) which shows that, on a risk-adjusted basis, the GA performed better than buy-and-hold despite negative absolute returns. In addition to that, the timing strategy yields superior average Sharpe ratios compared to the buy-and-hold benchmark throughout the entire 20-year data range (0.85 vs. 0.70). The authors further report that the amount of switches between asset classes is surprisingly low indicating that the GA picks up long term trends rather than reacting to short-term noise in the market[12]. As a by-product, this reduces total transaction costs which otherwise might cause a meltdown of excess returns generated by a timing strategy.

Neely (2003b) applies GP to the S&P 500 closely following the approach by Allen and Karjalainen (1999). The data range from 1929-1995. 5-year training periods were followed by a 2-year selection period. The best rules obtained were then tested out-of-sample on the remaining data. Including realistic transaction costs of 25 basis points, the author finds that GP generally underperforms a buy-and-hold strategy on a risk-adjusted basis. Therefore, he concludes that the EMH holds.

Setzkorn et al. (2003) use a GP framework just based on moving averages of various lengths to be determined by the GP algorithm to trade in the S&P 500. The approach features both a simple and a more complex setup. The data range from 1990-2001 on a daily basis and, as usual, was split up into training, validation and out-of-sample periods. Most notably, neither the single nor

[12]This might be one of the benefits from selecting macroeconomic variables rather than technical indicators as input to a GA/GP setup.

the complex setup succeeded in beating the buy-and-hold benchmark. Another noteworthy result is that the complex GP is found to be prone to overfitting resulting in a good fit during the training period and a poor fit in out-of-sample testing. In contrast to this, the simple algorithm performed worse during training, but better during out-of-sample than the complex algorithm. The authors consider the exclusive use of moving averages as indicators as the likely reason for the overall poor performance of their approach.

Potvin et al. (2004) apply a GP framework to the Toronto stock exchange. One of the special features of their study is that trading rules for fourteen individual stocks are derived rather than focusing on an entire index. The authors argue that this should allow for more individual and possibly more profitable trading rules. Furthermore, the methodology allows for the inclusion of short sales which would otherwise be difficult to implement when dealing with indices. Technical indicators like stock prices and trading volume are once more input to the algorithm; the data range from 1992-2000. The fourteen stocks were chosen to represent fourteen different industries in the TSE 300 index. In the end, GP underperforms the buy-and-hold benchmark on average. However, the stock-specific results are better; nine out of fourteen stocks show positive excess returns during the out-of-sample period. The overall poor performance is found to be caused only by a minority of stocks. Further analysis showed that when the benchmark buy-and-hold returns are close to zero or slightly negative, the GP-trading rules are profitable, implying a timing strategy to apply the trading rules when markets are stable or declining.

A more recent contribution is Fyfe et al. (2005), who apply and extend their framework (Fyfe et al., 1999) to the S&P 500, the S&P Auto and the S&P Bank index with data ranging from 1990-1999. In contrast to their previous study, they look for risk-adjusted excess returns. Although GP does find rules that easily outperform buy-and-hold, the picture changes after taking transaction costs into account and adjusting for risk. Under these restrictions, GP generally underperforms the appropriate buy-and-hold benchmark except for the S&P Auto where the algorithm partially outperforms the benchmark on a risk-adjusted basis. In defense of their study, the authors argue that their results might appeal to risk-seeking investors or investors with a context-dependent attitude towards risk[13].

Lipinski (2007) applies two refined GA to trade in stocks from the Paris stock exchange (in particular the automaker *Renault* stock) using data from 1999-2004. Training periods are 60 days followed by 20 days out-of-sample testing. The evolved trading rules beat the buy-and-hold benchmark regardless of which GA was used but the author finds that the more profitable algorithm also is more demanding in terms of CPU time.

Navet and Chen (2008) investigate GP performance on the New York stock exchange. Based on time series data of several stocks traded during 2000-2006, the authors explore the performance of GP trading rules based on a classification scheme distinguishing between stocks with high entropy and low entropy[14] using a variety of statistical techniques. The results are mixed with GP outperforming the benchmark for 3 out of 8 stocks. Interestingly, the authors find that GP performance, contrary to intuition, does not depend on the level of entropy ($\approx$ "predictability") of a stock and conclude that predictability is neither a necessary nor sufficient condition for profitability.

Apart from FOREX markets, Chen et al. (2008) also explore GP performance for eight stock markets (USA, UK, Canada, Germany, Spain, Japan, Taiwan, Singapore). The data cover 1989-2004 and are divided into rolling time frames of five years training followed by five years of validation and two years of out-of-sample testing. GP is found to consistently outperform buy-and-hold throughout all periods in the Tawainese market. In constrast, GP performance yields no outperformance in the other markets, among them the German DAX. The authors point out that the Taiwanese market has a quite different pattern compared to the other markets. In these markets, bull-markets are closely followed

[13] For example, the non-risk adjusted return of the GP-trading rules including 0.5% transaction costs for the S&P banks was 62.88% vs. just 20.72% for buy-and-hold during the 1995-1999 period.

[14] Loosely speaking a measure of future uncertainty of a dynamic process whose past is completely known.

by bear-markets which may lure GP into a buy-and-hold strategy during training and validation which eventually results in poor performance in the out-of-sample (bear-)market. As an interesting addition, the authors repeat their approach and allow for the possibility of short sales. However, GP performance does not improve in general. As another exercise, GP and buy-and-hold are compared with the performance of 21 human-generated trading rules. While these strategies generally underperform buy-and-hold and GP in bull markets, some of them manage to beat GP and buy-and-hold in all markets during a bearish period.

Drezewski and Sepielak (2008) focus on the Warsaw stock exchange for testing GP performance. Using data from 2001-2006 they find that GP outperforms buy-and-hold when applied out-of-sample to the same stocks that were used for training. In addition, the authors investigate how well GP is able to generalize beyond a selected stock by using a set of random stocks for training and a different set of random stocks for out-of-sample testing. However, the result are poor leading to the conclusion that GP fails to find general rules. As an interesting sidenote, the authors elaborate on GP convergence (i.e. GP fitness as a function of generations) and report that most of the fitness is achieved after rougly 25-75 generations (though they used 500 generations in total for each run) indicating that using more generations only results in very little additional fitness at the cost of dramatically increased CPU time.

For the sake of completeness, two more papers should be mentioned at this stage which are not based on "plain-vanilla" GA/GP-methodology but nevertheless share some common features with the studies reviewed so far. The first one is Yu et al. (2004) who use GP to find TTR for the S&P 500. They apply a refinement to the usual GP approach by using so-called λ-abstraction. Based on data ranging from 1982-2002, the standard GP (which is used as a comparison) is able to outperform the buy-and-hold benchmark; the λ-abstraction enhanced GP is able to improve upon the already positive results. The authors note that the outperformance is achieved in all market conditions which makes their approach a robust tool for profitable trading.

Another study worth mentioning is O'Neill et al. (2002) who, apart from the FTSE 100 and NIKKEI, also look at the German DAX. However, the approach is based on a different technique in evolutionary modeling called grammatical evolution which constitutes a class of its own. Therefore, the results are only partially comparable to the results of other studies presented in the literature review. Based on data ranging from 1991-1997 (DAX/NIKKEI) and 1984-1997 (FTSE 100), the performance of the approach is mixed. For the FTSE 100, the grammatical evolution technique slightly outperforms buy-and-hold while this benchmark is clearly surpassed in the case of the NIKKEI. Performance for the DAX is reportedly poor; the authors consider overfitting to be the likely reason for the poor results.

5.1.2 Foreign Exchange Markets

Neely et al. (1997) presented the first approach at using GP in FOREX markets. Six major exchange rates against the USD plus two cross-rates were subject of the study. The data ranged from 1974-1995. The training period for GP was 1975-1977, followed by a validation period from 1978-1980. Out-of-sample tests were conducted on the data for 1981-1995. Inputs for the GP setup were maxima, minima of prices, lagged prices, moving averages etc.. The algorithm was designed to yield simple buy- or sell-signals on a daily basis. The benchmark was simply zero return. The authors argue that a buy-and-hold strategy is not well-defined in FOREX markets since it always depends on the location of the investor whether she makes profits or not. For example, if the USD/EUR buy-and-hold return is positive for an U.S. investor, the converse is true for an European investor. Despite transaction costs, the authors find strong evidence of economically significant excess returns by using GP-evolved trading rules across the board with an overall average return for the out-of-sample period of 2.87% for all six currencies. Interestingly, the overall performance could be further improved by setting up a so-called median-rule portfolio, i. e. adopting an investment strategy that went long in a rate when more than 50 out of 100 rules turned out to be long in the market rather than just following the single best rule out of 100 rules generated per rate. The adoption of the median rule portfolio pushed average excess returns across all rates from 2.87% to 3.67%[15]. Although the authors stress that their generated rules are higly nested and complex, it turns out that one of the most profitable rules was as simple as “take a long position at time t if the minimum exchange rate over period $t-1$ and $t-2$ is greater than the 250-day moving average”. Further analysis of overall performance showed that the excess returns were not caused by implicit risk premia. In the end, the authors regard their findings as further evidence for inefficencies in the FOREX market.

In very much the same fashion, Neely and Weller (1999) shed further light on the power of GP trading rules in the FOREX market by extending existing analysis on the now defunct European Monetary System (EMS). Six European currencies against DM were subject of the study. The training period ranged from 1979-1983, validation period from 1983-1986 and the rules were tested out-of-sample from 1986-1996. Input to the GP setup were once more technical indicators. Mean excess returns from GP trading were found to be positive across the board (except for the DM/NGL rate), albeit not as high as in Neely et al. (1997). Average overall excess return was 1.62% which could be further improved by adopting the already mentioned median portfolio rule to 2.16%. The authors point out that the performance of GP had been probably dampened by the fixed rate bandwiths which were the most notable feature of the EMS. Further analysis showed that the excess returns could not be explained as compensation for higher risk. As a by-product, the trading rules were found to have some predictive ability in terms of market timing, i. e. when to be in-the-market and when to be out-of-the-market. In conclusion, the results for the EMS were in-line with the earlier findings for the USD-denominated market as shown in Neely et al. (1997).

Colin (2000) presents, in very much the same fashion as Colin (1994), a general framework for GP-assisted trading, this time with a real-world application to the FOREX market. A variety of technical trading indicators is used as in-

[15] While this still does not seem to be too much, it must be emphasized that the figures are average figures, shadowing the fact that the excess return in the USD/DM rate was in excess of 6%.

put for GP, among them oscillators, relative strength indices and directional movement indices. In total, Colin (2000) relies on seventeen different indicators popular among practicioners. Subject of the study are the USD/CHF and USD/JPY rates. The training period ranges from 1974-1981, validation period from 1981-1988 and test period from 1988-1995. By applying the best GP-generated trading rules, the author reports an average return of 6.5% and 7%, respectively for the two rates.

A noteworthy extension of the two contributions by Neely et al. just presented is Neely and Weller (2001). While the basic scope and setup largely remain the same, the GP setup is now provided with historical data on Federal Reserve interventions in FOREX markets to determine how excess returns found in previous papers relate to central bank action. Therefore, indicator variables signaling intervention via "buy USD", "no intervention", "sell USD" were added on top of the usual market data that served as input. The authors find some evidence of improved excess returns with monetary interventions for the US/GBP and US/CHF rates, but they also find that the positive impact declines over time. In contrast, the USD/DM and USD/JPY returns are even negatively affected by the inclusion of intervention data. Given the overall inconclusive results, the authors do not find evidence for the hypothesis that central bank intervention could be one of the causes for profitability of TTR. They argue that profitable trading is rather caused by strong and persistent trends in the FOREX market.

Dempster and Jones (2001) use technical indicators like moving averages and relative strength indices (six indicators in total). The intra-day data ranges from 1989-1996 on a 15-minute basis. Opposed to Neely et al. (1997) and Neely and Weller (1999, 2001, 2003a), the inputs for GP are based on a combination of existing, real-world indicators like the readily available 250-day moving average rather than letting GP derive artificial indicators. Furthermore, the setup allows for real two-way trading including short-selling instead of just determining whether to be in-the-market or out-of-the-market. The authors report mixed results. While they manage to find simple rules that earn up to 7% annually in the GBP/USD market at a statistically significant level, overall performance of the portfolio of trading rules is just about 5%. However, as they point out, their results are encouraging enough to justify further research.

Despite the (mostly) encouraging results, all three studies by Neely et al. share a common shortcoming, i. e. trading signals are based on daily data leading to highly unrealistic results such as trading frequencies ranging from once every two weeks to once in three months on average. As this phenomenon does not realistically reflect the speculative, fast-paced and higly liquid FOREX market, Neely and Weller (2003a) address this issue by applying their established framework to intra-day data and trading. In addition to the exchange rate and interest rate differentials, variables for the hour of the day were included in the input data for the GP setup as well. Training, selection and test periods were adjusted accordingly (2-months, 2-months, 7-months), the data are from 1996. As far as market quotes are concerned, half-hourly averages were used. As a result, the authors report that GP was not able to produce any excess returns for any currency considered (USD/DEM, USD/CHF, USD/JPY, USD/GBP) when taking realistic transaction costs into account. They argue that the surprising results which contradict the findings of their previous studies might be explained by the uneven division of capital allocated to trade in the FOREX market at different time horizons. They guess that most of the volume is generated by traders who close their position at the end of the day rather than investing with weekly or monthly horizons.

Austin et al. (2004) develop a GP intraday-trading framework for several currencies. Typical inputs include moving averages, stochastic oscillators, relative strength indices etc.. For the 1994-1998 period, trading is reported to be profitable out-of-sample after including realistic transaction costs. Annualized return for the GBP/USD was 13.07%, 4.29% for the USD/CHF and -0.37% for the USD/JPY rate[16].

Tsao and Chen (2004) take a theoretical approach by investigating the per-

[16] The authors do not elaborate further on their approach and results. Their system was developed in close collaboration with HSBC Global Markets and is therefore highly proprietary.

formance of GA for six different classes of time-series models, among them the classic ARMA, ARCH and GARCH processes. Rather than just testing GA on empirical data, the authors use Monte Carlo simulations based on these processes to evaluate the performance of GA vs. buy-and-hold taking into account returns, risk (Sharpe ratio), winning probability and a so-called luck-coefficent which loosely speaking tests whether an outperformance is based on just a few lucky trades. They find that GA performs particularly well in both linear- and nonlinear deterministic (chaotic) environments whereas they fail in nonlinear stochastic processes. As an empirical application, GA is tasked to evolve trading rules for EUR/USD and USD/JPY time series from January 1999 until April 1999. After establishing that the return series is fitted well by a mixture of MA(1) and GARCH processes (for which GA proved superior to buy-and-hold in the first part of the paper), GA is then shown to outperform the benchmark in terms of return, Sharpe ratio and winning probability.

The most recent contribution is Chen et al. (2008). They explore GP performance for eight major currencies (among them USD, DEM, JPY) using data from 1992-2004 divided into rolling 3:3:2 schemes, i.e. 3 years training plus 3 years validation period followed by a 2-year out-of-sample period. They report that GP generally fails at generating better returns than buy-and-hold. However, they extend their data and adapt their data division scheme to match the setup used in Neely et al. (1997) and Neely and Weller (1999) and find that GP is able to outperform the benchmark in 10 out of 12 scenarios at statistically significant levels. The authors conclude that the design of a data division scheme is paramount for GP performance. They particularly stress the importance of the length of a training period which should neither be too long nor too short for GP to pick up a pattern.

5.1.3 Futures and Bond Markets

GA/GP have found their way into futures markets as well. Wang (2000) applies GP to the S&P 500 futures market. Based on daily data and technical indicators such as moving averages, trading range breaks, volume etc. ranging from 1985-1998, the author picks GP-evolved trading rules based on 2-year training periods and applies them out-of-sample. The benchmark is a buy-and-hold strategy which consists of a long position in 2 S&P 500 futures contracts all the time. In contrast to this, the GP-based rules were designed to yield five different signals: long 2 contracts, long 1 contract, neutral (i. e. zero investment), short 1 contract, short 2 contracts (short-selling is assumed to be feasible, at least for institutional investors). Basically, Wang (2000) finds that GP-performance in the S&P 500 futures market is inconsistent. While some generated rules beat the benchmark, their overall power is limited often resulting in slightly negative excess returns when taking transaction costs into account. Overall performance seems to be better when markets are volatile whereas the GP-rules reportedly have difficulties in picking up sustained upward trends. Interestingly, although being unable to beat the benchmark, the GP-rules often converge to the buy-and-hold benchmark, i. e. 2 contracts long. The author notes that U.S. equity returns in the twentieth century (and therefore the higly-correlated futures markets) have been the highest of all countries making it difficult for GP to beat buy-and-hold.

In spirit of Allen and Karjalainen (1999), Karjalainen (2002) investigates the performance of a GP trading system for the S&P 500 futures market. The data range from 1982-1993. Inputs were moving averages, maximum/minimum of past prices, lagged prices etc.. The benchmark was once more a buy-and-hold strategy, i. e. a rolled over long position in a single S&P 500 futures contract. It turned out that the GP-based trading rules slightly outperform the benchmark for the 1988-1993 out-of-sample period. Further analysis showed that a portfolio of trading rules results in a superior annualized Sharpe ratio which is in-line with the findings in Allen and Karjalainen (1999) for the equity market. Therefore, GP-based timing strategies apparently reduce volatility by a significant amount while roughly matching buy-and-hold.

Tsang and Lajbcygier (2002) also explore evolutionary trading in futures. As a special feature, they make use of a standard GA and a Split Search GA[17].

[17] Basically, the idea is to start the GA search using two different input sets of variables for the starting solutions, one using variables from class x, the other using variables from class y. Solutions from the two separated evolutionary processes are allowed to eventually cross over in analogy to two slightly different species (say two different kinds of giant lizards on the

Basically, the data consist of daily highs and lows plus opening and closing prices for eight commodities between 1988 and 1998. The authors use a rolling-time frame of one year in-sample training followed by subsequent one year out-of-sample application throughout the entire data sample. The input data for GA consists of the classic filter rule (buy or sell contract when prices have decreased/increased by more than $x\%$) and a moving average filter rule which translates the classic filter rule concept to a smoothed time-series creating so-called percentage envelopes or volatility bands. Fitness of the trading rules is measured using the Sharpe ratio, benchmark strategy was buy-and-hold (only one contract long/short at any one time). The authors report that the Split Search GA and the standard GA only marginally beat the benchmark and that the results are not statistically significant. After changing the fitness measure to incorporate a take-profit mechanism plus the number of winning trades, the results improve but still lack statistical significance. Finally, the authors note that the difference in performance between the split island GA and the standard GA was almost statistically significant. In defense of their study, they point out that the primary goal was to show how GA performance can be improved by using modified GA and fitness functions that do not solely focus on total profitability.

Apart from the S&P 500, Bauer (1994) also looks at the U.S. government and corporate bond market. The approach is basically the same as already discussed in the section on equity markets. This time, the variables with the highest correlation with the spread between the long maturity Treasury bonds and the short maturity T-bills are the 1-month change in U.S. stock prices, economic growth momentum indicators, changes in unemployment levels and 3-month changes in consumer installment debt. The alternative investment to going long in Treasury bonds is going long in Treasury bills. In addition, a switching strategy between corporate- and Treasury bonds is considered using basically the same input variables. For the Treasury bonds/Treasury bills case study, the author reports an average return of the portfolio of GA-trading rules of 10.51% vs. 14.35% for buy-and-hold for the 1989-1991 out-of-sample period clearly miss-

Galapagos islands) living on two separate islands that sometimes, by crossing the water in between, manage to mate.

ing the benchmark. In contrast, the trading rules switching between corporate bonds and Treasury bonds slightly outperform the buy-and-hold benchmark for the same period (15.40% vs. 14.17%). As far as hedge portfolios are concerned (going long on a portfolio of good rules and going short on a portfolio of bad rules), the long portfolio partly outperforms the short portfolio in the case of Treasury bonds during the holdout period whereas the corporate bond hedge portfolio almost consistently outperformed the associated short portfolio. In conclusion, the Bauer (1994) approach seems to succeed in finding profitable Treasury/corporate bond market switching rules whereas the proposed Treasury Bills/Treasury Bonds switching rules perform poorly.

5.2 Hybrid Models

Figure 4.1 showcases the general framework of computer-aided trading systems with its main components, namely KBES, MTS, FS, ANN and GA/GP. A branch of the literature available focuses on combining two or more of these technologies, i.e. so-called hybrid models. Hybrid models are particularly popular in terms of GA/GP used to optimize ANN, nevertheless other hybrid approaches, such as a blend of GA/GP with FS, have been investigated as well with the main goal of merging the best of two worlds into a unified approach. A thorough discussion of the hybrid literature is beyond the scope of the thesis. Nevertheless, for the sake of completeness and for illustrative purposes, two hybrid models will be briefly reviewed to show how classic GA/GP-methodology can be extended to enhance performance of a trading system.

5.2.1 Neuro-Genetic Hybrid Models

A typical paper on this issue is Harland (1999). The main goal of his paper is to develop a hybrid Neuro-Genetic model for trading the US T-Bond future. As input, the author derived price transformations from underlying price data resulting in momentum indicators with different lags. Additionally, the data were enhanced by including similar momentum indicators from the positively correlated S&P 500 future. Next, the author set up a neural net based on these inputs and at this stage, GA come into play. With reference to Figure 4.2, a GA was used to run through a vast amount of possible neural net architectures in order to optimize the output (i. e. convergence on the training data set). This was achieved by determining the optimal number of hidden layers of the ANN by using GA. The results indicate that the final model mostly outperforms a roll-over buy-and-hold strategy in T-Bond futures during the out-of-sample test period. The amount of winning trades is reported to be significantly higher than 50%.

A similar approach is presented in Kwon and Moon (2003), who apply a Neuro-Genetic framework to trade various stocks on the New York stock exchange and NASDAQ. The genetically evolved neuronal networks are able to outperform buy-and-hold. In addition, a comparison between GA-evolved ANN and stand-alone ANN shows that the use of GA considerably enhances the performance of neural nets. However, neither Harland (1999) nor Kwon and Moon (2003) investigate the question whether neuro-genetic hybrids deliver better performance than GA/GP-only based approaches.

Azzini and Tettamanzi (2008) also use a neuro-genetic hybrid to find profitable trading strategies for the stock of Italian car company *Fiat* using closing prices, moving averages and other technical indicators ranging from 2003 until 2006 on the Milan stock exchange. They find that their models yield significant excess returns on a risk-adjusted basis.

5.2.2 Fuzzy-Genetic Hybrid Models

A Fuzzy-Genetic hybrid is disscussed in Lam et al. (2002). They pick typical TTR and fuzzify them which translates into rules such as "if the relative strength index is *high*, then buy". Profitable combinations of these rules were then evolved using a GA. Applied to five different stocks that trade on the NASDAQ, the Fuzzy-GA hybrid is found to indicate the right market-timing signal in 68.3% of cases and yields an average return of 11.1%. As an interesting extension, the algorithm is redesigned to incorporate feedback from profitability during out-of-sample testing to evaluate whether the system needs to be recalibrated by retraining. The dynamic retraining approach leads to improved profits of 51.5% with 67.8% of trades being profitable.

A similar approach is discussed in Costa Pereira and Tettamanzi (2006). Using open, high, low, closing prices, exponential moving averages and stochastic oscillators, suitable intra-day trading rules are evolved using a Split Search GA as presented in Tsang and Lajbcygier (2002). The results are then fuzzified into human-interpretable rules. The trading rules are applied to the Dow Jones, Nikkei 225 and single stocks in different markets. Based on 2-,3- and 5-years of training (2002-2006 maximum), out-of-sample (first half of 2007) GA performance is superior to buy-and-hold for the Dow Jones both in terms of excess return and risk-adjusted return as indicated by the Sharpe ratio whereas the approach fails to outperform the benchmark in the Nikkei 225 and the single stocks considered.

5.3 Evolutionary Modeling in Forecasting

At this stage, some comments on GA/GP-based forecasting are in order. During compilation of the literature review, it turned out that forecasting is to a certain degree intertwined with AA/TS. A suitable forecasting system may be exploited to form the basis of a profitable trading system. Furthermore, the relation sometimes works vice-versa as well. For instance, Pereira (2002) notes that the trading rules found for the Australian stock exchange apparently possess some forecasting power since, based on relevant technical market data, they suggest long positions when volatility of returns is low and stay out-of-the-market when volatility of returns is high. Similar findings have been reported by Allen and Karjalainen (1999). As a rule of thumb, forecasting models can be distinguished from AA/TS by the fact that they rather explore the correctness of their predictions instead of exhaustingly exploring profitability. Therefore, in a strict sense, the already presented paper by Frick et al. (1996) could be considered to fall into this category as well. For the sake of completeness and to "get the big picture", forecasting papers based on GA/GP will be briefly reviewed to outline the links between GA/GP-based AA/TS and forecasting.

The first contributions to the issue are Chen and Yeh (1996, 1997) who emphasize the distinction between predictability and profitability when testing the

EMH. They investigate the forecasting power of GP-based models of stock returns against the random walk hypothesis using subsets of the S&P 500 and the Tawainese stock exchange TAIEX ranging from 1971 to 1994. Based on the function set and inclusion of constants and lagged returns as sole inputs, the resulting GP forecasts are (non-)linear autoregressive models[18]. The authors find that using larger training samples tend to make the EMH seem valid as GP forecasts largely fail at beating the random walk hypothesis. Interestingly, the random walk hypothesis, i.e. $E(r_t|\Omega_{t-1}) = 0$ where r_t denotes the return today based on yesterday's information set Ω_{t-1}), is proposed by GP several times as the best forecasting rule. Things look different when using shorter training samples as GP tends to forecast returns better than the random walk hypothesis. The authors note that the forecasts get better when using more generations during the GP evolution process thus increasing CPU time=search cost. Chen and Yeh conclude that even though the EMH sometimes may not be 100% valid, it might be too costly (from a computing/search cost of view) to exploit possible weaknesses.

Another forecasting paper closely related to the preceeding discussion is Li and Tsang (1999) who use a GP approach to forecast whether it is possible to achieve a return greater than r within the next n trading days in the Dow Jones index. Input to the algorithm were once more technical indicators (moving averages, trading rule breaks, filter rules). The authors use a standard GP design for searching over a large set of trading rules. Based on $r = 2.2\%$, $n = 21$, the authors find that their methodology yields better forecasts than a purely random alternative. The results are generally confirmed in a similarly designed follow-up study (Li and Tsang, 2002).

Another example illustrating a forecasting approach is Kaboudan (2000, 2002). The most notable difference to the trading system papers presented so far is that the author does not derive trading strategies directly using GP, but rather uses GP to evolve suitable regression models that forecast stock prices. Technical indicators were used as input to the GP regression model evolver with the

[18] The linear case may emerge as well of course if considered fit enough by GP.

main goal of forecasting one-day-ahead intra-day high and lows for four stocks from the New York stock exchange and NASDAQ composite. Compared to a naive forecast (tomorrow's high and low equal today's high and low), the GP approach is found to perform slightly better, with accuracy oscillating around 55%.

As already pointed out, though forecasting and AA/TS are to some extent intertwined, GA/GP in forecasting applications constitute a class of its own within research directed at evolutionary modeling of financial markets. Therefore, a thorough discussion of the contributions published so far is beyond the scope of this thesis.

Now that applications of EA in AA/TS have been reviewed, it is time to take a closer look at the inner workings of GP.

Part III

The Mechanics of Genetic Programming

6 Introductory Remarks

The main goal of this chapter is to explain the mechanics of GP prior to presenting real-world case studies on testing stock market efficiency using GP. The chapter is organized as follows. The next section gives a brief account of the development of GP from a historical point of view, followed by a brief discussion of the strengths and weaknesses of GP methodology. The basics of GP are then thoroughly discussed which constitutes the main part of the chapter followed by some remarks on parameter choice which is paramount to successful application of GP. The next section aims at providing more insight into the question as to why GP works from a theoretical point of view. The chapter ends with some concluding remarks.

7 Historical Overview

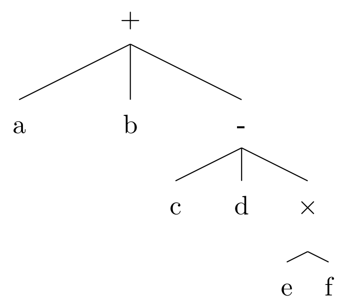
GP belongs to a field called Evolutionary Computing (EC) in computer science[19]. EC comprises different optimization techniques that all share the common theme of emulating evolutionary processes found in Nature. Early attempts at implementing this idea date back to the mid-sixties with the works of Fogel et al. (1965, 1966) who proposed a technique termed Evolutionary Programming[20]. Later, in the mid-seventies, Holland (1975) introduced the concept of GA. At about the same time, Rechenberg (1973) developed a similar technique called Evolutionary Strategies. Two decades later, Koza (1992) invented GP. Koza's main intention was to create a framework for self-programming computers. With the help of GP, program code could be expressed as a hierarchical tree-like structure originally encoded in the LISP (LISt Processor) programming

[19]Note that EC and the term EA are interchangeable.
[20]The basics of Evolutionary Programming are explained in Eiben and Smith (2003).

language which was subject to genetically inspired changes. LISP uses a special syntax which makes encoding of trees particularly easy. Basically, commands are entered by appropriate operators and arguments which are coded in parentheses and evaluated from the innermost term to the outermost term with the operator read first followed by the appropriate arguments. For example, the LISP expression

```
(+ a b (- c d (× e f)))
```

translates into



in terms of GP visualization.

Since Koza's seminal contribution, several sub-variants of the basic GP methodology have been proposed which are outlined in Banzhaf et al. (1998) and Nedjah et al. (2006). Throughout the remainder of the thesis, the basic GP proposed by Koza (1992) as presented in the upcoming discussion will be used. A more detailed account of the different flavours of EC is given in Eiben and Smith (2003) and Banzhaf et al. (1998), respectively.

8 Why use Genetic Programming?

For the application at hand, the major reason for applying GP to a finance-related problem is basically a one-liner: GP (and EA in general) is potentially able do deal with a wide range of problems including optimization of nonlinear processes (Koza, 1992; Keane, 2001)[21]. GP requires very little in terms of input

[21] For some real-world applications, see Cantú-Paz et al. (2003) for example.

to solve an optimization task and builds, or better said, evolves potentially suitable computer programs (=trading rules) to solve a given problem. There is considerable evidence that financial markets are highly nonlinear which will be elaborated upon in the next subsection.

8.1 Financial Markets and Nonlinear Dynamics

One of the major topics in finance is the analysis of asset returns as a building block for modeling financial markets with the ultimate aim of deriving profitable investment strategies with as-low-as-possible risk. Doing so requires qualification and quantification of the economic variables that inherently drive financial markets and a suitable modeling method. A technique for modeling markets are linear regression models. Among this class are standard econometric models like *AR(p)* and *MA(q)* processes and, as a blend of the latter two, *ARMA (p,q)* models. Although well-established and quite popular, the linear approach lacks explanatory power too often which is illustrated in Franses and van Dijk (2000). Therefore, alternative and possibly more sophisticated techniques are needed to capture the complex dynamics of financial markets.

There is considerable evidence that markets are driven by nonlinear dynamics. Several tests have been developed capable of detecting nonlinearities in time series data such as Tsay (1986) and Brock et al. (1987). For example, Scheinkman and LeBaron (1989) employ the so-called BDS-Statistics derived in Brock et al. (1987) to test weekly return data from 1928-1985 of the CRSP (Center for Research in Securities Prices) value-weighted U.S. stock index. The analysis is extended in Brock et al. (1991) to include S&P 500 data for the same period. The authors find evidence of nonlinearities for both indices. Hsieh (1989) checks exchange rates for nonlinearities. The data consist of daily closing prices from 1974-1983 for the Canadian Dollar, Deutsche Mark, Japanese Yen and Swiss Franc each quoted in U.S. Dollar. The test results suggest strong nonlinearities in the data across all five currencies. Franses and van Dijk (2000) make use of an ANN to detect nonlinearities and find evidence across all major currencies as well. They extend their approach to include international stock markets and find evidence of nonlinear dynamics across the globe, namely for the Frankfurt, Paris, London, New York, Hong Kong, Singapore and Tokyo stock market indices.

As far as modeling nonlinearities is concerned, the classic *ARCH (p,q)* (Engle, 1982) and *GARCH (p,q)* (Bollerslev, 1986) models are probably the most widely used technique. In addition, many other approaches with different levels of sophistication and requirements in terms of their suitability to model an underlying process are available. An overview of alternative techniques is given in Tong (1990). Returning to a more general level, the line of reasoning for applying GP to financial markets is as follows: Markets are driven by nonlinear processes and GP is in theory capable of dealing with nonlinearities. Therefore, the idea is to let GP evolve trading rules in a financial market and check whether the results comply with the EMH.

8.2 General Properties of Genetic Programming

The discussion so far focused on the link between GP and financial markets. On a more general level, some points can be made as to what makes GP a promising algorithm for problem solving. According to Keane (2001), EC-based techniques and thus GP have the following advantages over more traditional techniques:

- *Applicable to a wide range of problems*
- *Low development and application cost*
- *Easily incorporated into other methods (hybridization)*
- *Solutions are interpretable*
- *Can be run interactively and allow for incorporation of user-proposed solutions*
- *Often provides many alternative solutions.*

In contrast, some of the drawbacks are:

- *No guarantee for optimal solution within finite time*

- *Weak theoretical basis*
- *May need parameter tuning for good performance*
- *Often computationally intensive and thus slow.*

As far as the pros are concerned, the wide applicability of EC regardless of the underlying process and the interpretability of solutions are by far the most attractive properties, especially in comparison to neural nets which are often powerful but difficult to interpret due to the "black box" property. Concerning real-world applications in finance, Bauer (1994) points out that GA are well suited for easily checking functional relationships between economic variables without resorting to complicated yet often unsatisfying econometric techniques. In terms of easy hybridization, the papers by Harland (1999), Lam et al. (2002) and Kwon and Moon (2003) demonstrated how GP/GA can be merged with other techniques to form a unified approach.

A serious drawback is the lack of theoretical foundation which particularly affects GP whereas the theory behind GA is somewhat better understood[22]. This issue will be addressed in more detail later. Another serious issue is the lack of general rules for parameter choice and the associated fact that the impact of parameter choice on GP results is not well understood, at least in finance applications (Navet and Chen, 2007).

Having casually outlined the genesis and reasons for using GP, it is now time for a more detailed approach. Before doing so, it must be emphasized that the following topics are fairly standard material that is extensively covered in the original source Koza (1992) and Banzhaf et al. (1998). These references provide an in-depth discussion of GP while the following basic definitions are meant to help understand real-world GP applications, i.e. testing stock market efficiency.

[22] In modern GA theory, GA are interpreted as Markov processes which considerably facilitates derivation of analytical results, see Reeves and Rowe (2003) for details.

9 The Basics of Genetic Programming

9.1 GP-Parameters for Tree Phenotypes

As already pointed out before, GP-based solution candidates for an optimization problem are encoded in a hierachical tree-like structure. The size, shape and contents of these trees is controlled by a variety of parameters and sets which are next on the agenda.

9.1.1 Terminal Set

*The **terminal set** $\mathcal{T}$ is the set of all inputs to a GP system including constants and zero argument functions.* Casually speaking, $\mathcal{T}$ defines the contents of the leaves of a GP-tree.

9.1.2 Function Set

*The **function set** $\mathcal{F}$ consists of the statements, operators and functions available for a GP system.*

A wide range of functions is possible such as

- Boolean operators: `AND, OR, NOT, XOR`[23]
- Comparison operators: $\leq$, $\geq$, $<$, $>$
- Arithmetic functions: `+, -,` $\times$, $\div$
- Mathematical functions: `sin, cos, exp, log, sqrt`
- Conditional statements: `IF, THEN, ELSE, CASE, SWITCH.`

Further functions such as loops (while...do etc.) and variable assignment functions are available as well. Generally speaking, there are virtually no limits to functions for GP, i. e. almost any function can be used. This allows for extreme flexibility in terms of the solution found by GP. Nevertheless, it is crucial for function sets that they comply with the so-called *closure property*. The closure property states that all functions should be able to deal with any constant

[23] eXclusive OR, or casually speaking "either of the two, but not both".

from the terminal set and any value returned by a function from the function set, otherwise the GP run could fail. Examples are the well-known "divide by zero"-error or the log of a negative number which, depending on function and terminal set, may occur during a GP run. Fortunately, the toolkit GPLAB running on Matlab which will be used throughout the thesis automatically takes care of the closure requirement[24].

Another important concept is the *sufficiency property.* The sufficiency property states that the terminal set and the function set should be selected in such a way that GP potentially can find an acceptable solution. For example, it is highly unlikely that GP will capture an even trivial nonlinear relationship if it is only based on the spartanic function set $\mathcal{F}_i = \{+, -, \times\}$. Extending $\mathcal{F}_i$ to $\mathcal{F}_j = \{+, -, \times, \div, \exp, \log, sqrt\}$ will likely result in a higher chance of finding a nonlinear relationship. However, a too complex function and/or terminal set might result in an extremely large search space leading to poor results as well. Koza (1992) points out that in the end, it is up to the user to decide whether the sufficiency property is met based on the individual optimization task at hand.

9.1.3 (Maximum)-Depth

The ***depth*** *of a node is defined as the minimal number of nodes that must be traversed to get from the root node of the tree to the selected node. The closely related* ***maximum depth*** *refers to the largest distance between the root node and the outermost terminals.* Broadly speaking, the maximum depth defines the size/complexity of solution candidates.

9.1.4 How to grow Trees

Combining the sets and parameters just discussed, the question remains how to initialize tree-based GP solutions in the very first generation G_{t_0}. Koza (1992) distinguishes between two methods called *grow* and *full.* The *grow* method randomly selects nodes from both the function and terminal set. The only ex-

[24] As an illustrative example, this can be achieved by making use of a so-called protected division operator that simply returns one if division by zero would occur otherwise. To guard against the log of a negative number, the absolute value of the argument is taken to prevent the GP run from crashing. Other potential violations of the closure property are handled by similar special operators during runtime. See Silva (2007) for details.

ception is the root node which is exclusively based on the function set. When a branch reaches a terminal node it is cut off even if the maximum depth would allow for another level of branching. Therefore, the *grow* method produces trees of irregular shape.

In contrast, the *full* method selects nodes from the function set only until the maximum depth has been reached. Then it selects terminals only. As a result, every tree branch has the full maximum depth, trees are regularly shaped. Koza (1992) suggests using the so-called *ramped half-and-half method*, i. e. 50% of the initial generation uses grow and the other 50% uses the full method in order to ensure a wide genetic variety of solutions to start with.

Having outlined the basic ingredients for determining tree phenotypes, it is now time to give an account of the parameters that control evolutionary dynamics.

9.2 Genetic Operators

9.2.1 Crossover

The crossover operator controls swapping of genetic material between two individual trees (parents). Two parents are chosen from an initial population based on their respective fitness[25]. Once two parent trees have been selected, a random subtree in either parent tree is selected. They are then swapped between the two parents resulting in two children. The process is illustrated in Figure 9.1.

Crossover is probably the most important operator in GP. By splitting and mixing (already promising) parental genes, crossover is in theory assumed to breed children solutions with improved fitness. However, there is considerable debate as to whether potential solutions really profit from crossover. While Koza (1992) argues that crossover likely preserves good solutions and builds even better solutions, Banzhaf et al. (1998) question the overall beneficial effect of crossover. They argue that crossover does not distinguish between good and bad building blocks thus potentially ripping apart promising subtrees of a parent solution. In a standard regression application, Nordin et al. (1995) and Nordin and Banzhaf

[25]See subsection 9.3.

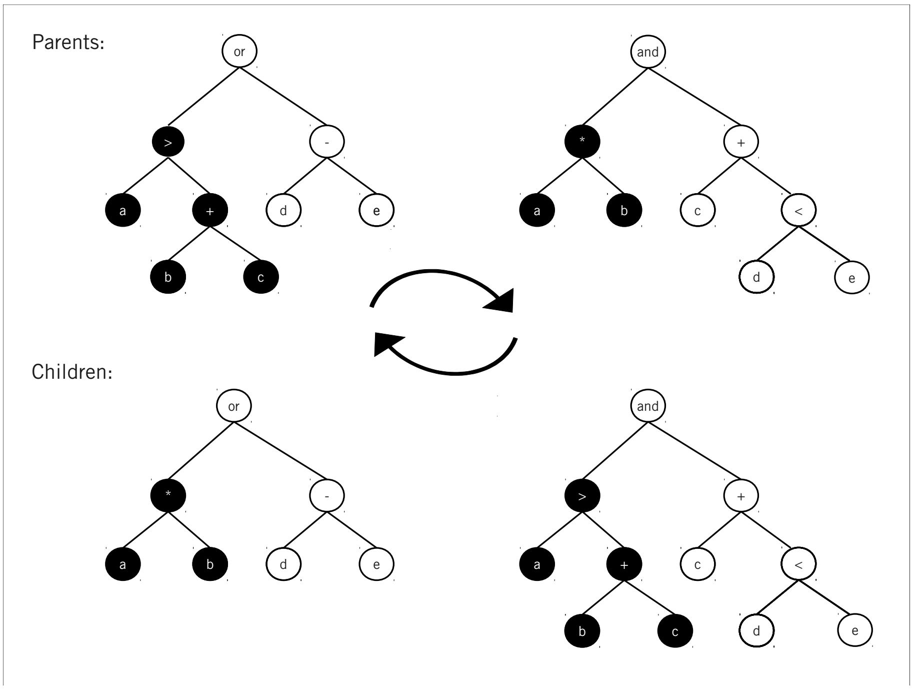
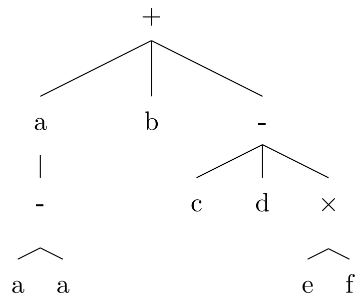


Figure 9.1: Example of crossover: A random subtree (shaded nodes) in either parent is randomly selected and swapped resulting in two children.

(1995) report that crossover is potentially lethal to good solutions most of the time or at most neutral and only rarely improves fitness between parental and children generations. However, they also find that the overall negative effect can be partially reversed in later generations within a GP run due to the build-up of what they call introns or bloat. Basically, bloat is a phenomenon that appears in late GP runs. With more and more generations, solutions tend to gain complexity although their overall fitness *can* but need not necessarily increase. Bloat (or introns) are referred to as code within a solution that is somewhat superfluous and does not affect in any way, positive or negative, fitness. For example, the tree



features bloat, that is the subtree `(- a a)` . The point is that bloat "distracts" the crossover operator from ripping apart potentially good subtrees such as the (assumedly) powerful subtree `(- c d (×  e f))` since the probability of being cut off by crossover is equal for all nodes within a solution tree. Therefore, the individual probability for each node of being cut off by crossover is inversely related to the presence of bloat within solutions. Nordin et al. (1995) distinguish between a structural and a global protection role of bloat . The former allows a population to protect highly fit building-blocks, the latter protects an individual solution almost completely against a destructive crossover.

However, the authors only draw their conclusions from a standard regression application and crossover might have a different impact in other applications. Furthermore, due to technical reasons, the authors only use a very limited function set. Whether the overall impact of crossover is benefical, negative or neutral remains an open question in theoretical GP[26]. More elaborate versions of crossover exist which are discussed in Banzhaf et al. (1998). However, only the basic crossover operation will be used throughout the thesis.

9.2.2 Mutation

Mutation is one of the key aspects in genetics and therefore also plays an important role in GP. First, a parent tree is picked by the algorithm based on its respective fitness. Then a node or terminal in the parent tree is randomly selected with equal probability, cut off at the respective branch and replaced with a newly generated random subtree which complies with the depth and size

[26] See section 10.2.1 and 10.2.2.

parameters for the respective GP run in order to avoid excessive bloat[27]. After this procedure, the child tree is inserted into the new population. Mutation basically aims at introducing (hopefully promising) new genetic material into the gene pool.

9.2.3 Reproduction

Reproduction is the easiest operator in GP. It is asexual, meaning that a member of the population is chosen based on fitness and copied unchanged into the next generation[28]. Reproduction constitutes so-to-say a safe haven for (already good) solutions to be carried over to the next generation since the competing crossover operator must not necessarily yield better offspring. Therefore, reproduction ensures that a pre-defined proportion of good genes is passed on to the next generation without suffering from the effects of crossover and mutation.

More genetic operators like permutation, editing and encapsulation are described in Koza (1992) and Banzhaf et al. (1998). However, these operators will not play a role in the subsequent discussion.

9.3 Fitness Function and Selection

Fitness is defined as a measure of how well a solution candidate is adapted to the environment. The primary purpose of fitness within GP is to determine the quality of a solution in order to assign individual probabilites for passing on genes to the next generation. Fitter candidates should be allowed to live on with a higher probability than less fit solutions. The selection algorithm determines the way an individual is selected for crossover, mutation and reproduction. A variety of selection schemes exist with so-called fitness-proportional selection being the most popular. Given a population of n individuals with respective fitness $f(i)\ \forall\ i = 1, ..., n$, the probability p_i for individual i to pass on its traits to the next generation (via crossover, mutation or reproduction) under fitness-proportional selection is given as

[27] The newly generated subtree is created again by using either the *full*, *grow* or *ramped half-and-half* method. See section 9.1.4.

[28] Similarly to cloning in genetics. For some reason, the term reproduction rather than cloning is used in GP literature.

$$p_i = \frac{f(i)}{\sum_{i=1}^{n} f(i)}. \quad (9.1)$$

Another important selection algorithm is the so-called tournament selection. Rather than evaluating an entire population, a subset of the population is randomly chosen. The member solutions contained in this subset then compete against each other. The better solutions are cleared for reproduction with mutation and replace the worse solutions. They are then inserted back in the population. By working with subsets of populations, tournament selection saves a considerable amount of CPU time which is why it has become a popular tool in GP. When using tournament selection, an additional parameter for controlling tournament size t_s has to be set. A small t_s results in a low selection pressure, a high t_s in a higher selection pressure. This potentially helps to quickly achieve convergence (that is, no better solutions can be found in further generations) within a GP run. However, only fitness-proportional selection will be used throughout the thesis.

9.4 Parameter Choice

According to Koza (1992), the most important parameters are the population size $\mathcal{M}$ and the maximum number of generations $\mathcal{G}_{max}$. Depending on the complexity of the problem to be solved, higher values for these parameter tend to yield better solutions at the cost of increased CPU time. As much of GP is based on heuristics, there is no general rule as to how to optimally set parameters. Instead, Koza (1992) suggests a rule-of-thumb approach that has shown decent performance across a variety of applications from different fields[29]. In GP literature, the so-called Koza tableau is an established way of presenting a particular GP setup. An example is given in Table 9.1.

9.5 A Basic GP Run

Summarizing the discussion so far, a basic GP run involves the following steps:

[29] In Koza (1992), applications range from artifical ants which, by means of GP-optimized movements, gather a maximum of food in their habitat to strategic decision making for game-theoretic applications.

Population size $\mathcal{M}$	500
Maximum number of generations $\mathcal{G}_{max}$	51
Probability of reproduction p_r	0.1
Probability of crossover p_c	0.9
Probability of mutation p_m	0.1
Initial population initialization: Ramped half-and-half	
Selection algorithm: Fitness-proportional	

Table 9.1: A basic Koza tableau. The odd value for $\mathcal{G}_{max}$ stems from the intial generation G_{t_0} plus fifty subsequent generations.

1. *Define terminal and function set*
2. *Define fitness function and associated selection algorithm*
3. *Choose parameters (population size, maximum number of generations, maximum depth, crossover/mutation/reproduction probability, termination criterion etc.).*

GP is a so-called generational EA, i. e. GP distinguishes between well-defined, discrete generations $\mathcal{G}_{t_0}$, $\mathcal{G}_{t_1}$, $\mathcal{G}_{t_2}$...$\mathcal{G}_{t_n}$=$\mathcal{G}_{max}$. Generation $\mathcal{G}_{t_i}$ is created from $\mathcal{G}_{t_{i-1}}$ for $i = 1, ..., n$ and replaces it completely. The basic dynamics of GP is illustrated in Figure 9.2. Once the initial generation has been created in a random fashion, all individuals are measured in terms of fitness and the fittest individuals are subject to genetic operators. Once the next generation is fully populated, the algorithm checks whether the maximum number of generations has been reached. If not, the new generation completely replaces the old generation and the new individuals are once more measured in terms of fitness and are subject to crossover, mutation and reproduction until the next generation is fully populated and so on. If the termination criterion is met (maximum number of generations reached), then the best individual from the final generation is the result of the GP run.

To bridge the gap between the rather generic description of EC/GP and the practical application of testing stock market efficiency throughout the remainder of the thesis, some of the terms just presented can be translated from the theoretical realm into more applied terms for real-world applications inspired by Keane (2001) as follows:

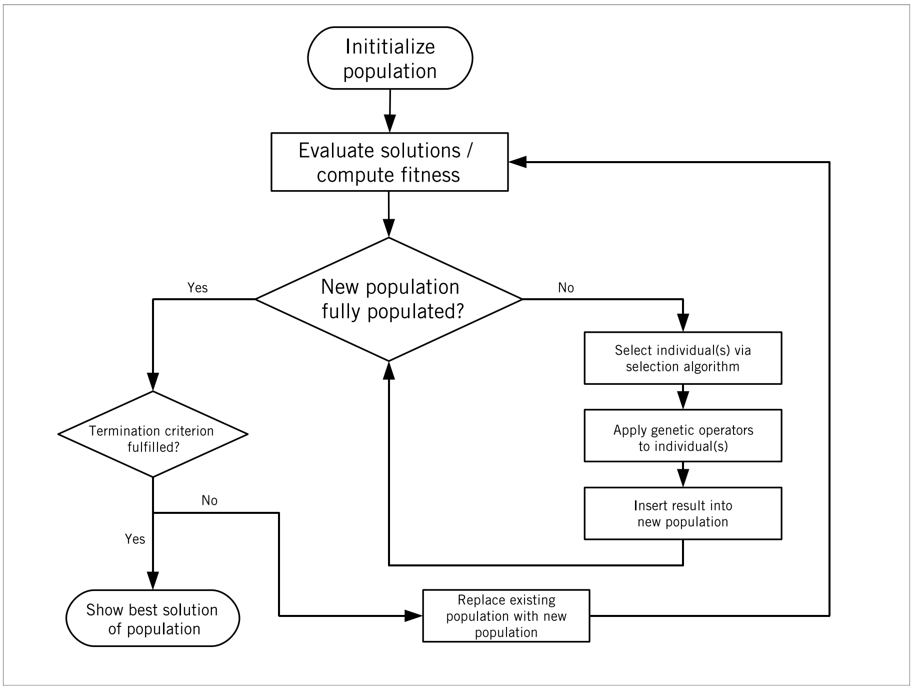


Figure 9.2: *A basic Genetic Programming flowchart.*

Evolution		*Problem Solving*		*Real-world Application*
Individual	⟷	Candidate Solution	⟷	Trading Rule
Fitness	⟷	Quality	⟷	Excess Return
Environment	⟷	Problem	⟷	Stock Market

Therefore, abstract terms like "individual" can be thought of as one (out of infinitely many) trading rules whose quality/fitness (i. e. excess returns) is evaluated in order to determine individual survivability in the environment which is the stock market.

10 Why does Genetic Programming work?

Having discussed various operators and parameters in GP, the question arises as to why GP is suitable for a broad range of optimization tasks. The main result of modern GP theory is that the power of GP is based on so-called schema theory, which broadly speaking describes how various combinations of genes evolve throughout a GP run. The upcoming discussion aims at shedding some light at schema theory in an informal way. A more rigorous and comprehensive account of schema theory is given in Langdon and Poli (2002). But prior to a more concise discussion of schema theory, a very general result from population genetics will be presented first.

10.1 Prize's Theorem

An important analytical result from population genetics was proposed by Prize (1970) and reformulated by Langdon and Poli (2002) for use with GP. Basically, the theorem relates the change in frequency of a gene within a population from generation G_{t_i} to $G_{t_{i+1}}$ to the covariance between individual fitness (=number of offspring) and the frequency of a given gene in G_{t_i} as follows.

$$\Delta Q = \frac{cov(z,q)}{\bar{z}} \cdot \frac{\sum z_i \Delta q_i}{M\bar{z}} \tag{10.1}$$

where:

Q	$=$	Frequency of a given gene/linear combination of genes in the population
ΔQ	$=$	Change in Q from one generation to the next
q_i	$=$	Frequency of the gene in individual i
Δq_i	$=$	Change in frequency of the gene in individual i
z_i	$=$	Number of offspring produced by individual i ($\approx$ fitness of individual i)
$\bar{z}$	$=$	Mean number of children produced.
M	$=$	Size of initial population

(10.1) holds for a single gene or any linear function of any number of genes. According to Prize (1970), the term at the right in (10.1) cancels out on average, i. e.

$$E\left[\sum z_i \Delta q_i\right] = 0 \tag{10.2}$$

and can therefore be omitted. As a matter of personal taste, equation (10.1) can be written in a slightly more intuitive way by substituting $cov(z,q)$ and considering (10.2) which yields

$$\Delta Q = \frac{\rho_{zq} \cdot \sigma_q \cdot \sigma_z}{\bar{z}} \tag{10.3}$$

The most striking feature in (10.3) is the correlation coefficient ρ_{zq} which translates into

$$\begin{array}{ll} \Delta Q \uparrow & 0 < \rho_{zq} \leq 1 \\ \Delta Q = 0 & \rho_{zq} = 0 \\ \Delta Q \downarrow & 0 > \rho_{zq} \geq -1 \end{array}$$

Therefore, if there is a positive relationship between the mean number of children (i. e. fitness) and the presence of a particular gene or linear combination of genes, the respective genetic material will spread further in future generations whereas the frequency remains unchanged if $\rho_{zq} = 0$. The respective gene/genes might become extinct in the long run if $0 > \rho_{zq} \geq -1$.

Alternatively, Prize prefers to recast equation (10.1) in terms of a linear regression model by substituting $cov(z, q)$ which results in

$$\Delta Q = \frac{cov(z, q)}{\bar{z}} = \beta_{zq} \frac{\sigma_q^2}{\bar{z}} \tag{10.4}$$

with the usual interpretation for the slope coefficient β_{zq} from econometrics. Interestingly, Prize's Theorem implicitely considers fitness to be the only factor affecting gene frequency within a population as, due to (10.2), the effects of crossover and mutation have no significant impact on average. Despite this point of view, Altenberg (1994) and Langdon and Poli (2002) argue that the theorem can be applied to GP as well. However, albeit in some highly artifical and complex scenarios, Langdon and Poli (2002) show that (10.2) does not always hold.

10.2 Schema Theory and Building Block Hypothesis

The basic idea as to why GP works is based on schema theory. A schema is a similarity template that encompasses certain compononents, or to put it in Koza's (1992) words: "...the set of all individual trees from a population that contain, as subtrees, one or more specified subtrees. That is, a schema is a set...sharing common features". For example, the schema $\mathcal{H}$=[(+ x y), (× 2 x)] stands for all trees that include at least one occurence of the subtree (+ x y) or at least one occurence of the subtree (× 2 x). Under the assumption that trees that contain the schema $\mathcal{H}$ have on average higher fitness, schema theory aims at analyzing how a schema propagates from generation to generation within a GP run under the effects of selection, recombination, crossover and mutation. As a more universal definiton of a schema $\mathcal{H}$, the "#"-character which stands for "don't care" can be used as shown in O'Reilly (1995) and O'Reilly and Oppacher (1995). For example, the schema $\mathcal{H}$=[(× # x)] stands for all trees that include any element from the function set $\mathcal{F}$ or the terminal set $\mathcal{T}$ multiplied by x. Therefore, the schema $\mathcal{H}$ can be matched several times within a single program. Several schema theorems have been proposed out of which two proposals will be discussed to give an idea how GP is able to find solutions in the search space.

10.2.1 Koza's Schema Theorem

The first, albeit informal approach at schema theory was proposed by Koza (1992). He argues that programs containing good schemata have on average higher fitness values than competing programs within the same generation. Higher fitness results in a higher probability of reproduction. Following this line of reasoning, good schemata will live on and will be combined by the crossover operator to even better schemata. As it is more likely that crossover disrupts a complex schema, small schemata will profit from crossover; good but complex schemata are likely to get disrupted by crossover. This leads to the evolution of small but powerful schemata throughout the generations of a GP run which are then combined by crossover to even better solutions. Over time, this leads GP to search more promising parts of the solution space and, to put it in Koza's words "...concentrates the search of the solution space into subspaces of LISP S-expressions of ever-decreasing dimensionality and ever-inreasing fitness". The final solution is then evolved as a blend of various small but powerful schemata which is known as the so-called building block hypothesis. Interestingly, Koza does not further comment on the mutation operator and considers crossover to be the only operator that powers GP. Consequently, he uses $p_m = 0$ in his real-world applications. As another issue, Koza considers the crossover operator to preserve good, albeit small, schemata. This point of view is highly contested. Banzhaf et al. (1998) point out the ambiguities of the crossover operator which, due to space constraints, would be inappropriate to discuss here further.

10.2.2 O'Reilly's Schema Theorem

Koza's work was formalized by O'Reilly (1995) and O'Reilly and Oppacher (1995). For fitness-proportionate selection and the special case of $p_m = 0$ (no mutation), the following can be shown to hold.

$$E[i(\mathcal{H}, t+1)] \geq i(\mathcal{H}, t) \cdot \frac{f(\mathcal{H}, t)}{\bar{f}(t)} \cdot \left[1 - p_c \cdot \overbrace{\max_{h \in Pop(t)} P_d(\mathcal{H}, h, t)}^{P_d(\mathcal{H}, t)} \right] \tag{10.5}$$

where $i(\mathcal{H}, t+1)$ denotes the number of instances of a schema $\mathcal{H}$ in generation $t+1$, $i(\mathcal{H}, t)$ the number of instances of schema $\mathcal{H}$ in generation t, $f(\mathcal{H}, t)$ the mean fitness of all instances of $\mathcal{H}$[30] and $\bar{f}(t)$ the average fitness in generation t. The constant p_c stands for crossover probability and $P_d(\mathcal{H}, h, t)$ for the probability of destruction of schema $\mathcal{H}$ in program h in generation t due to crossover[31]. As $P_d(\mathcal{H}, h, t)$ varies between different programs (=tree-encoded solutions) within the same generation, the authors decided to make use of a maximum operator which results in expression (10.5). However, this causes estimates of $i(\mathcal{H}, t+1)$ to be very conservative as criticized by Banzhaf et al. (1998) and Langdon and Poli (2002). Furthermore, $P_d(\mathcal{H}, t)$ varies from generation to generation. O'Reilly and Oppacher (1995) admit that due to the variability of $P_d(\mathcal{H}, t)$, no real hypotheses about the propagation and creation of building blocks can be made. Consequently, the question as to whether crossover has on average a destructive or preserving effect on building blocks remains open[32]which illustrates the often criticized weak theoretical foundation of GP. Furthermore, O'Reilly's approach does not consider the effect of mutation which would complicate analysis even more.

[30] This can be computed as the weighted sum of the fitnesses of the solutions that match $\mathcal{H}$, using as weights the ratios between the number of instances of $\mathcal{H}$ that each program contains and the total number of instances of $\mathcal{H}$ in the population.

[31] This is defined as the ratio between the number of links in the tree fragments plus the number of links connecting them of $\mathcal{H}$ in h and the total number of crossover locations in h.

[32] In contrast to Koza (1992) who argues that crossover generally preserves good schemata. A detailed discussion of the ambiguous role of crossover in GP can be found in Banzhaf et al. (1998).

10.2.3 Other Schema Theorems

Other schema theorems apart from O'Reilly (1995) and O'Reilly and Oppacher (1995) have been proposed. An analytical result for the frequency of a program in $\mathcal{G}_{t+1}$ was first proposed by Altenberg (1994), followed by Whigham (1995, 1996a, 1996b), Rosca (1997) and Langdon and Poli (2002).

The difficulty with all the approaches just mentioned is that, although still being analytically tractable, they exhibit considerable complexity which mirrors the complexity of the underlying GP structure, particularly the ever changing shape of solution trees due to the effects of crossover and mutation. Consequently, the terms derived are quite complex and little intuitive. An in-depth discussion of these results is given in Langdon and Poli (2002).

10.2.4 Criticisms of Schema Theorems

Schema theorems can be criticized on grounds of being of little use in practical GP applications. As seen in (10.5) for example, schema theorems usually only work with expected values which, at the end of a GP run, makes it hard to use a theorem recursively to predict GP behaviour from generation to generation[33]. Furthermore, schema theorems only give lower bounds rather than exact results. However, Langdon and Poli (2002) overcome these weaknesses at the expense of tractability. They find some support for the building block hypothesis but also stress that building blocks need not necessarily be of the short, low-order and highly fit type.

10.2.5 Genetic Programming vs. Random Search

The discussion so far has pointed out the inherent evolutionary dynamics within GP that pushes the algorithm to find better solutions from generation to generation. This makes it highly unlikely that GP solutions are just results of blind random search. Koza (1992) provides several informal arguments against the blind random search thesis by stressing the fact that GP usually starts with a very low fitness in the inital generation[34] and then improves fitness throughout

[33] Unless one assumes the population to be infinite which is unhelpful in real-world applications.
[34] Which, by the way, is simple random search unlike the following generations.

the generations often yielding very good solutions at the end of a run. He argues that this process from zero to surprisingly good solutions alone is proof that GP is not blind random search. From a more empircal point of view, Koza (1992) also runs a number of experiments pitting GP against blind random search using up to 10 million random solutions and finds that random search is in neither case superior to GP. Rather the opposite is true with GP beating random search very clearly in all experiments conducted. He concludes that with GP being superior to blind random search, it is highly unlikely that GP just comes up with solutions that could have been found by blind random search as well.

10.3 Concluding Remarks

For the remainder of the thesis suffice it to say that the informal building block hypothesis brought forward by Koza (1992), despite some arguable weaknesses[35] and side-by-side with the universally applicable Prize's Theorem has some explanatory power as to how GP seeks the solution space and is able to find near-optimal solutions to optimization problems. The chapter demonstrates that GP, while being a suitable and often powerful optimization technique as seen in the second chapter of the thesis, arguably has a weak theoretical basis compared to other established techniques.

[35] See Nordin et al. (1995a, 1995b), Banzhaf et al. (1998) and Langdon and Poli (2002).

Part IV

Testing Stock Market Efficiency via Genetic Programming

11 Introduction

After the literature review in the second chapter and the discussion of the inner workings of GP in the third chapter it is now time to apply GP to test stock market efficiency. The basic outline of the chapter is as follows. The next section briefly reviews the definition and implications of market efficiency followed by a brief account of market efficiency tests that have been used in the past. The following section is the nucleus of the thesis and presents the technical setup and test results for two stock markets, namely the German DAX and Hong Kong's main index, the Hang Seng. After an extensive discussion of the results obtained, the chapter concludes with some final remarks on market efficiency from a GP point of view.

12 Some brief Remarks on Market Efficiency

First of all it must be emphasized that the upcoming discussion does not even attempt at giving a comprehensive account of the efficient market literature. It is safe-to-say that research on this issue is abundant and Fama (1970) already points out that it is difficult to do justice to all contributions published so far. Unsurprisingly, the task has not become any easier almost forty years later to put it mildly. The abundance of research is mirrored in a dedicated category in the *Journal of Economic Literature* (JEL) classification scheme[36]. Consequently, and to avoid losing focus, only the landmark survey articles on market efficiency will be considered with the ultimate goal of setting the stage for the author's personal contribution to the issue and to show how the GP-based approach fits into the big picture.

[36] JEL Code G14.

12.1 Definition and Implications of the Efficient Markets Hypothesis

To make the EMH work, two assumptions are necessary (Fama, 1970):

1. Market equilibrium can be expressed in terms of expected returns.
2. The set of all information available at time t concerning a security, ϕ_t, is fully exploited by market participants for formation of expected returns.

Adopting the setup from Fama (1970), condition (1) can be formally expressed as

$$E(\tilde{p}_{i,t+1}|\phi_t) = [1 + E(\tilde{r}_{i,t+1}|\phi_t)]\, p_{it} \tag{12.1}$$

where $\tilde{p}_{i,t+1}$ denotes the price of security i (which is a random variable as indicated by the tilde) at time $t+1$, ϕ_t the set of all information available at time t concerning a particular security, $\tilde{r}_{i,t+1}$ the return of security i at time $t+1$ (once more a random variable) and p_{it} stands for the price of security i at time t. The assumption that market equilibrium can be expressed in terms of expected returns implies that returns in excess of the equilibrium expected returns should be inexistent provided ϕ_t is fully exploited by the market participants. Formally, the following is assumed to hold. Defining the excess returns of security i at time $t+1$ as

$$x_{i,t+1} = r_{i,t+1} - E(\tilde{r}_{i,t+1}|\phi_t) \tag{12.2}$$

where $r_{i,t+1}$ denotes the observed return at $r+1$, the EMH implies that

$$E(\tilde{x}_{i,t+1}|\phi_t) = 0 \tag{12.3}$$

i. e. the martingale property must hold. Particularly important for the thesis at hand is the special case

$$E(\tilde{p}_{i,t+1}|\phi_t) \geq p_{it} \tag{12.4}$$

or alternatively

$$E(\tilde{r}_{i,t+1}|\phi_t) \geq 0 \tag{12.5}$$

in (12.1) which, in case of a strict inequality, denotes the submartingale property. Therefore, prices may increase but not in excess of expected equilibrium prices at $t + 1$. This implies that abnormal returns (returns in excess of equilibrium expected returns) cannot be achieved by *any* trading system.

The EMH can be divided into three broad categories which in turn can be verified empirically (Fama, 1970; Jensen, 1978):

1. **Weak efficiency:** The information set ϕ_t only contains the past prices of a security up to time t
2. **Semi-strong efficiency:** All past prices plus all other publicly available information up to time t are contained in the information set ϕ_t
3. **Strong efficiency:** All information, including insider information up to time t make up the information set ϕ_t.

With the inclusion of risk-adjustment and transaction costs, a more refined version of the EMH was formulated by Jensen (1978) who stated that "a market is efficient with respect to the information set ϕ_t if it is impossible to make economic profits by trading on the basis of information set ϕ_t", where the term "economic profits" means risk-adjusted returns net of all costs.

Basically, the thesis revolves around this particular issue. Under the EMH, no trading system should be able to beat a simple buy-and-hold strategy in the same security (or index as will be the case in the subsequent analysis). If a GP-powered trading system defies the implications stated above after risk-adjustment and inclusion of appropriate transaction costs, the validity of the EMH may become questionable, at least during certain periods of time in a security market. In the thesis at hand, it is up to GP to find such trading systems, provided they exist at all.

Another important concept in connection with the EMH is the random walk hypothesis first brought forward by Bachelier (1900) and Samuelson (1965).

Basically, if prices follow a random walk, price changes should be white noise which is in line with the EMH. Therefore, the random walk hypothesis implicitely addresses EMH issues as well so as a by-product, the random walk hypothesis will be tested as well to some degree in the thesis at hand.

12.2 Techniques for Testing Market Efficiency

As pointed out in the introduction of the chapter, the author does not even remotely attempt at giving a review of EMH literature. Instead, only the basic test techniques will be briefly discussed in order to see how GP fits into the picture. Fama (1991) proposes three categories for EMH tests, namely

- tests for return predictability (weak efficiency)
- event studies (semi-strong efficiency)
- tests for private information (strong efficiency).

The empirical literature published so far mainly focuses on the first and second EMH category (weak and semi-strong efficiency) whereas the third (strong efficiency), which is a rather strong assumption, has only been rarely tested. The thesis itself can be categorized under the weak efficiency tests as only input data based on closing prices will be used.

As Fama (1970) points out, any empirical test for efficiency requires a definition of the information set ϕ_t. It is clear that there is no general consensus as to what constitutes ϕ_t. By picking a reasonable selection of variables, only one out of infinitely many information sets can be used to test for market efficiency[37]. As a workaround, tests have to be based on a proxy set of information that most investors agree on. This applies to both semi-strong and strong-form efficiency tests. In contrast, weak-form tests are quite easy to implement as price/return data are clearly defined and readily available at low cost to virtually every market participant. Weak and semi-strong empirical EMH tests may be based on (in no particular order):

[37] For example, mostly economic variables should be contained in ϕ_t whereas some investors might prefer the inclusion of somewhat far-fetched variables such as the frequency of sunspots or the amount of rain on Wall Street. Therefore, the one and only ϕ_t does not exist.

- Statistical tests of indepedence/tests of white noise (random walk) (Fama, 1965)
- Return predictability tests/seasonal anomalies tests (Fama and French, 1988)
- Event studies (Miller and Reilly, 1987)
- Direct tests of trading rules/trading systems (Alexander, 1961; Brock et al., 1992; Ratner and Leal, 1999)
- Volatility tests (Shiller, 1979)
- Cointegration tests/tests for bubbles (Islam and Watanapalachaikul, 2005)

whereas strong-form empirical EMH tests mainly rely on tests of private, i.e. insider information. Agents having some sort of insider information concerning a security may be corporate insiders, stock exchange specialists, stock analysts and money managers. As a classic example, Niederhoffer and Osborne (1966) focus on the informational advantage of stock exchange specialists. Other examples of strong-form tests include Scholes (1972), Jaffe (1974) and Seyhun (1986).

As can be seen in the enumeration above, the thesis at hand being based on computer-generated and GP-optimized trading rules fits into the category of direct tests of trading rules/systems pioneered by Alexander (1961)[38]. The list above is by no means exhaustive, however most of the existing literature on the EMH fits into one of the categories just mentioned. A detailed account of EMH-tests can be found in Fama (1970, 1991) and Bollerslev and Hodrick (1994).

Summarizing the discussion and considering Islam and Watanapalchaikul (2005), major challenges to the EMH arise from

- empirical tests whose results do not support EMH
- shortcomings in statistical and mathematical modelling

[38]It should be noted that the exisiting GP/GA literature as discussed in the second chapter fits into this category as well. Another sub-category of trading system tests is the abundant literature on neural networks. This gives just a glimpse of the vast empirical EMH literature.

- predictability of excess volatility and mean reversion
- speculative bubbles
- complex nonlinear dynamics in financial markets.

The last item is particularly important for the thesis as GP has been mainly designed for nonlinear optimization. Having roughly outlined the fields of EMH research, it is now time to set the stage for a GP-based approach to testing the EMH.

13 Testing Stock Market Efficiency via Genetic Programming

13.1 Introduction

The upcoming discussion is made up of two subsections. The first deals with implementation issues and describes the setup necessary to find trading rules using GP. The remaining subsection is the core of this chapter and the thesis presenting, analyzing and discussing the out-of-sample results obtained from running the GP algorithm for two stock markets, namely the DAX and the Hang Seng. The choice of the DAX and Hang Seng is motivated by the lack of GP-related research on these markets. Additionally, it is tempting to compare the results obtained as the two markets fall into different categories: The highly liquid and well-established DAX which is more likely to be efficient vs. a highly volatile emerging market where subtle inefficiencies may be, if they exist at all, easier to find. It is up to GP to uncover possible subtle inefficiencies by deriving optimized trading rules.

13.2 Implementation

13.2.1 Technical Aspects

In order to implement a GP algorithm, the first issue is the choice of a suitable software framework. The original implementation as proposed by Koza (1992) is based on the LISP language which is one of the oldest high-level programming languages available. Generally speaking, a GP framework can be implemented using any advanced programming language such as the aforementioned LISP, Java, Fortran, C and C++ to name a few. However, the author came across a toolkit called GPLAB (*G*enetic *P*rogramming *Lab*oratory) (Silva, 2007) based on Matlab[39], an industry standard for technical calculations and programming language with emphasis on vector and matrix computation. GPLAB is open-source under the GPL-license[40] and is maintained by Sara Silva of the Evolutionary and Complex Systems Group at the University of Coimbra, Portugal. Due to its computer science related background, the toolkit basically incorporates just two classic GP applications first discussed in Koza (1992), that is symbolic regression[41] and the so-called artificial ant problem[42].

Therefore, it was necessary to adapt GPLAB to deal with financial applications. Due to the open-source nature of the toolkit, appropriate changes to the source code could be made to accommodate the "breeding" of trading rules with their respective performance calculations. Prior to elaborating on the implementation details of the algorithm, a look at the data used for the study is next on the agenda.

13.2.2 Data+Stylized Facts

The data used throughout the thesis were obtained from *Yahoo! Finance* (DAX + Hang Seng closing prices), Deutsche Bundesbank (FIBOR/EURIBOR rates) and the Hong Kong Monetary Authority (HIBOR[43]). All data are on a daily basis. The stock index data are in either case closing prices adjusted for splits and dividends. Money market rates are those reported at 11:00 a.m. on each respective trading day. The total data sample used ranges from 1997-2007 for both indices. For obvious reasons the author refrained from considering the

[39]The MathWorks, http://www.mathworks.com/products/matlab/.

[40]Which basically means that the program is non-commerical and that the source code is freely available. The source code may be altered and redistributed as long as the changes to the original code are documented. For details, see http://www.gnu.org/copyleft/GPl.html.

[41]Meaning that GP is tasked to find a suitable regression function given a set of points in $\Re^2$.

[42]The artificial ant can be thought of as a robot that has to search for food pellets spread across a chessboard-like surface. By finding rules such as "if food-pellet ahead move forward else turn left" GP tries to evolve movement rules for the robot which maximize the amount of food pellets found.

[43]Hong Kong Interbank Offered Rate.

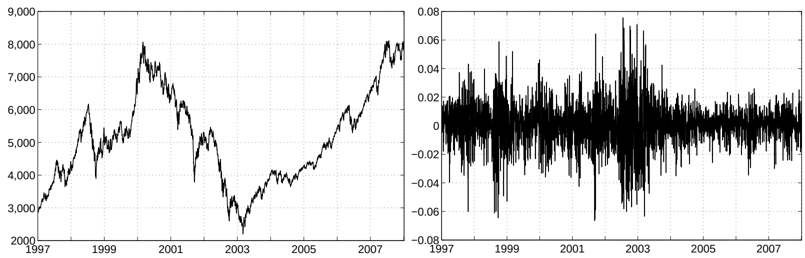


Figure 13.1: *DAX time series data from January 1997 - December 2007 and associated log-returns.*

2008 data as well which mark a major structural break in markets worldwide. The remainder of the thesis is based on the aforementioned 11-year sample.

To get a feel for the data, Figures 13.1 and 13.2 display time series charts of closing prices together with the associated daily log returns. The DAX starts at about 2800 points in 1997 and peaks at more than 8000 points in 2000 fueled by the new economy bubble. Soon after, the bubble collapses with the DAX reaching a low of about 2200 points in 2003. After that, a steady recovery leads the index north up to 8000 points at the end of 2007. Interestingly, despite a couple of swings beforehand probably caused by a partly collapse of the new economy bubble, volatility in the DAX increases considerably after Sept. 11th 2001 and it is not until 2004 that the index gets into calmer water.

Starting at 13700 points, the Hang Seng is soon hit by the 1997 Asian crisis reaching a low at 6500 points in late 1998. The index recovers over the subsequent years and slowly declines once more with the trough in early 2003. Since then, the index has recovered in a sustained fashion hitting a high of about 31000 points in late 2007. Most of the volatility present in the Hang Seng occurs as part of the Asian financial crisis in 1997 whereas Sept. 11th while still being notable, has surprisingly little impact compared to the aforementioned event. In recent years, volatility has been lower[44].

To gain some more insight into daily log returns, summary statistics are provided in Table 13.1. Mean daily returns in the DAX have been positive and higher

[44]This might be considered proof of a maturing market.

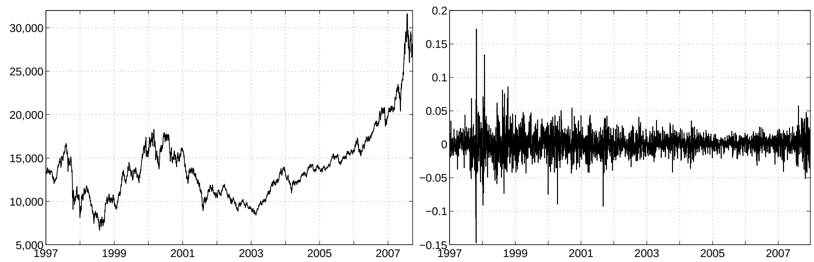


Figure 13.2: *Hang Seng time series data from January 1997 - December 2007 and associated log-returns.*

than in the Hang Seng during the sample period. The Hang Seng features a maximum daily return of 17% along with a spectacular minimum of -14%, both due to the Asian crisis. The standard deviation is considerably higher than in the DAX[45]. The time series are skewed slightly to the left (DAX) and slightly to the right (Hang Seng). The excess kurtosis shows that either time series is highly leptocurtic with excess kurtosis in the Hang Seng being roughly four times higher than that of the DAX[46].

	DAX	*Hang Seng*
Sample	1997-2007	1997-2007
# Observations	2784	2711
Mean	0.0003738	0.0002689
Median	0.001061	0.0005427
Minimum	-0.06652	-0.1473
Maximum	0.07552	0.1725
Std.Dev.	0.01555	0.01698
Skewness	-0.1521	0.1298
Ex.Kurtosis	2.4162	10.3544

Table 13.1: Summary statistics for daily returns.

13.2.3 Trading Rule Inputs

As a quick reminder, the GP-based EMH tests in this study are based on so-called endogenous variables, meaning closing prices and derivatives thereof. For

[45] A visual comparison between Figure 13.1 right and 13.2 right is misleading due to different y-axis scale used for the DAX and Hang Seng.

[46] This adds up to the perception of the Hang Seng as an emerging market index.

the application at hand, rather traditional and basic input variables have been chosen. All inputs are based on end-of-day closing prices adjusted for splits and dividends. Figures 13.1 and 13.2 show some degree of non-stationarity, particulary for the Hang Seng[47]. As this may be harmful to GP performance, the data have been normalized by dividing each closing price by its respective 250-day moving average which is the standard procedure used in Neely et al. (1997), Neely and Weller (1999) and Allen and Karjalainen (1999)[48]. Since this would mean the loss of approximately one year consisting of roughly 250 trading days for either data sample, data from 1996 have been added to compute normalized data for the core input sample 1997-2007. After normalization, the data hover around 1. As most of the data are between 0.8 and 1.2, constants in steps of 0.01 within this interval have been chosen as additional terminals[49] making rules such as $X_1 < 1.04$ possible. A refined input with smaller terminal steps and/or a wider interval for terminals would not necessarily result in better rules by spanning a solution space that may be too big for GP to perform satisfactory (Koza, 1992). Despite normalization of closing prices, all return calculations are based on non-normalized data. In addition to closing prices, moving averages, max- and minima and lagged variables are available as input as well. All three indicators are derived from the normalized closing price series. The moving average indicator of length θ at time t is defined as:

$$MA_t(\theta) = \frac{1}{\theta}\sum_{i=0}^{\theta-1} P_{t-i} \qquad \forall \theta \in \{1, 2, 3, ...\}. \tag{13.1}$$

Maxima and minima over different time frames θ are implemented as:

$$P_t^{max}(\theta) = Max\,[P_{t-1}, ..., P_{t-\theta}] \tag{13.2}$$

$$P_t^{min}(\theta) = Min\,[P_{t-1}, ..., P_{t-\theta}] \tag{13.3}$$

[47] For a formal investigation, a unit-root test would be necessary. The author is aware of this but deliberately chose to skip the test in order to avoid losing focus. Even if the hypothesis of a unit-root would be rejected for the DAX, it would not seem sensible to conduct further GP studies with non-normalized data for the DAX and normalized data for the Hang Seng (which clearly is non-stationary).

[48] In fact, the use of non-normalized data has been shown to degrade GP performance (Chen et al., 2008).

[49] See chapter 9.1.1.

$$\forall\theta \in \{1, 2, 3, ...\}.$$

Last but not least, lagged prices of P_t are defined as

$$P_t^{lag}(\theta) = [P_{t-\theta}] \qquad \forall\theta \in \{1, 2, 3, ...\} \tag{13.4}$$

where θ indicates the length of the time lag. To ensure a sufficient variety of short- to long-term time frames, all indicators above have been implemented with

$$\theta \in \{3, 5, 10, 15, 30, 50, 100, 150, 200, 250\}$$

with θ counted in days. The choice of indicators in conjunction with different time frames results in 41 indicators available as input to GP trading rules[50]. A (very simple) trading rule could look like: *if*

$$MA_t(100) < P_t^{max}(30) \tag{13.5}$$

then buy the index or *if*

$$P_t > P_{t-150} \tag{13.6}$$

then buy the index, else stay out and earn the risk-free return on the money market[51].

13.2.4 Fitness Function

Focusing on implementation again, the algorithm runs over the training sample first and creates suitable trading rules. Two asset classes are available for investment, either a long position in the index or an investment in the money market[52][53]. When a rule has been created, the rule loops over all trading days (that is over all rows of the respective input matrix) of the training sample to

[50] $10 \times MA_t(\theta) + 10 \times P_t^{max}(\theta) + 10 \times P_t^{min}(\theta) + 10 \times P_t^{lag}(\theta) + 1 \times P_t$, the normalized closing price series=41 variables. For a typical 3-years training period in the DAX, the resulting matrix is of dimension (759×41).

[51] Money market rates are not entirely risk-free due to counterparty risk, however they may be considered a proxy for the risk-free rate.

[52] Note that short positions are not allowed to avoid complications.

[53] Since a long position in stocks covers the whole respective index, the asset allocation problem could be restated as "when to go long in ETF". ETF are a very recent innovation in financial markets so the author does not consider them explicitely and instead assumes "classical" trading with higher transaction costs than ETF investments would imply.

determine whether the particular rule is true or false for a particular trading day. The result is a binary matrix with dimensions (1×759) for 3-years training and (1×1265) for 5-years training with "0" indicating an out-of-the-market position for a particular trading day and "1" indicating an in-the-market position[54].

As all calculations are based on end-of-day closing prices, the position suggested by the GP trading rule is entered into the following day. This introduces a so-called slippage error. Slippage means that the very first quote the next day is not necessarily equal to the closing price the day before. However, this problem which is also pointed out in Dempster and Jones (2001) is dealt with by using a conservative estimate of transaction costs. Therefore, part of this slippage is accounted for in a relatively high transaction cost (25 and 50 basis points).

Focusing again on the binary matrix indicating which position to take each day, a zero position for a particular day triggers a FIBOR/EURIBOR or HIBOR investment the next day. The daily return from an out-of-the-market position equals

$$r_f(t) = log\frac{(1 + r_{f,monthly})}{\delta} \tag{13.7}$$

with $r_{f,monthly}$ denoting the prevailing 1-month money market rate. δ indicates the mean number of trading days per year. As seen in Table 13.1, the total number of trading days for the DAX over 11 years is 2785 which equals $\delta = 253$ trading days per year on average. Things look slightly different for the Hang Seng. As the total number of trading days is just 2723, the respective mean is 247 so that in the Hang Seng case, the equation above is used with $\delta = 247$ trading days.

The shortest money market rate is the 1-month interbank rate in both markets and will be used throughout the thesis. All calculations are based on business days rather than actual calendar days. Therefore, in case of a prolonged out-of-

[54]Matrix dimensions are based on the assumption that the mean number of trading days in the DAX is 253. For the Hang Seng, the mean number of trading days is 247 so the respective matrices become slightly shorter.

market position, the code checks whether an out-position exists for more than 22 business days which approximately equals a calendar month with 30 days. If so, the investment has reached maturity and if an out-position has to be prolonged further, a rollover occurs adapting the respective new 1-month rate. This rule of thumb introduces a marginal error since EURIBOR and HIBOR are calculated as $\frac{actual}{360}$. It might happen that a prolonged out-position actually misses the correct revolving date since 22 business days may mean a 30-days+ position in actual calendar terms due to weekends and national holidays. The consideration of business days only has been chosen to avoid the tedious exercise of determing weekends and national holidays for the extensive data samples used in this study. As daily variances of interbank rates are quite low compared to stock prices, the error from missing the correct rollover date by 1- or 2 days may be considered as marginal. Last but not least, the use of $\delta = 247, 253$ rather than $\delta = 360$ in (13.7) at first sight overstates the return from a money-market position but the effect should even out since only business days earn interest rather than calendar days. The difficulties of calculating appropriate daily risk-free rates from longer maturities are discussed in Vaihekoski (2009).

To sum up the discussion, calculation of daily risk-free rates is not abolutely precise but the total effect on GP returns should be marginal. Some authors of GP studies do not even include the possibility for earning a risk-free rate at all (see Chen et al., 2002)[55].

If an in-the-market position is indicated by GP, the return π_i (open position at b_i, close, sell at s_i) including transaction costs c is calculated as (Allen and Karjalainen, 1999):

[55]The author of the thesis had an informal discussion with a senior fund manager during which the manager pointed out that money market returns are so low compared to a strategy that is in the market at the right time for just a couple of days that the money market could be safely excluded from the study. Nevertheless, the author decided to include at least a proxy for the risk-free rate.

$$\pi_i = \frac{P_{s_i}}{P_{b_i}} \cdot \frac{1-c}{1+c} - 1 = exp\left[\sum_{t=b_i+1}^{s_i} r_t\right] \cdot \frac{1-c}{1+c} - 1 \tag{13.8}$$

$$= exp\left[\sum_{t=b_i+1}^{s_i} r_t + log\frac{1-c}{1+c}\right] - 1 \tag{13.9}$$

with

$$r_t = log\left(\frac{P_t}{P_{t-1}}\right). \tag{13.10}$$

A position is either 100% in-the-market (long position in the index) or 100% out-of-the-market (earning the respective money market rate), i. e.:

$$I_b(t) \times I_s(t) = 0 \quad \forall t \tag{13.11}$$

with

$$I_b(t) = \begin{cases} 0, & \text{if } I_s(t) = 1 \\ 1, & \text{if } I_s(t) = 0 \end{cases}$$

or

$$I_s(t) = \begin{cases} 0, & \text{if } I_b(t) = 1 \\ 1, & \text{if } I_b(t) = 0 \end{cases}$$

equivalently, where $I_b(t), I_s(t)$ denote indicator functions. If an in-position is held at the very last trading day, the position is forcibly closed.

The total performance of a trading rule is then computed as

$$r_{gp} = \sum_{t=1}^{T} r_t I_s(t) + \sum_{t=1}^{T} r_f(t) I_b(t) + n \cdot log\frac{1-c}{1+c} \tag{13.12}$$

with n denoting the total number of trades (open/close long position in index). The simple return is defined as

$$\pi = e^{r_{gp}} - 1 \tag{13.13}$$

and the return from a buy-and-hold strategy over the respective period is

$$r_{bh} = \sum_{t=1}^{T} r_t + log\frac{1-c}{1+c}. \tag{13.14}$$

Then the fitness=excess return of a trading rule is calculated as

$$\Delta r = r_{gp} - r_{bh}. \tag{13.15}$$

The evolutionary process of creating a rule and evaluating its performance as described in equations (13.8)-(13.15) is repeated over and over again until the number of individuals and number of generations requirement of the GP algorithm has been met[56].

Trading rules should be robust against overfitting. This is sometimes difficult as the training data may contain some noise patterns that might get picked up by the GP algorithm. However, finding technical trading rules is based on the assumption that there are some regularities in the underlying data. Telling these apart from noise patterns is one of the tasks GP has to deal with. Ideally, GP should be able to generalize trading rules beyond the training sample. Therefore, the data samples used in this study are divided into training and subsequent out-of-sample periods.

The fittest trading rule from in-sample is applied out-of-sample. Return calculation is the same as in-sample except that only the best trading rule found during training is applied out-of-sample[57].

It is paramount to the analysis that excess returns still persist after appropriate risk adjustment. As a measure of return to risk, the Sortino ratio (Sortino and Price, 1994) will be used throughout the thesis. The original implementation of the GP algorithm included the classic Sharpe ratio (Sharpe, 1966) as risk-adjusted performance measure. However, it turned out during evaluation of the trading rules that returns frequently did not meet the implicit requirement of

[56] The process is depicted in Figure 9.2.

[57] Therefore, out-of-sample calculations are straightforward in contrast to the complex evolutionary breeding of rules during in-sample training.

being normally distributed. Rules that spent prolonged periods of time out-of-the-market were particularly affected by this since they tend to gather a lot of very small absolute returns from money market investments while incurring substantial losses during a few in-the-market days (excess kurtosis). This pattern led to overly optimistic Sharpe ratios and consequently, the Sharpe ratio approach had to be abandoned in order to avoid biased results.

The Sortino ratio is a variation of the Sharpe ratio that is more robust when dealing with skewed distributions (Chaudry and Johnson, 2008). As a distinct feature, it only considers downside deviation (DD) (Sortino and van der Meer, 1991) that occurs when returns fall below the so-called minimum acceptable rate of return (MAR). When returns exceed the MAR, the respective upside risk is not taken into account which is intuitive from an investor's point of view. Therefore, downside risk is penalized whereas upside risk is neutral within the Sortino ratio framework[58]. Adopting the notation from Chaudry and Johnson (2008), the Sortino ratio can be formally defined as follows. The mth order lower partial moment is defined as:

$$LPM = \frac{1}{N}\sum_{t=1}^{N}(r_t - L)^m I(r_t \leq L), \tag{13.16}$$

with

$$I = \begin{cases} 0, & \text{if } r_t > L \\ 1, & \text{if } r_t \leq L \end{cases}$$

where L is some threshold, N the number of returns and r_t the return at time t. (13.16) is a general risk measure and can be adapted for a number of special cases with one of them being DD. Setting $m = 2$ and $L = MAR$ in (13.16) yields the downside deviation:

$$DD^2 = \frac{1}{N}\sum_{t=1}^{N}(r_t - MAR)^2 I(r_t \leq MAR) \tag{13.17}$$

[58] As a sidenote, Chen et al. (2008) suggest to take downside risk into account in future GP studies.

with

$$I = \begin{cases} 0, & \text{if } r_t > MAR \\ 1, & \text{if } r_t \leq MAR \end{cases}$$

Next is the choice of a suitable MAR. As the Sortino ratio serves as a performance measurement geared towards evaluating mutual fund performance, a natural choice would be the risk-free rate. Alternatively, a benchmark rate like the return on an index could be chosen. However, in the thesis at hand, MAR has been set to zero. This is mainly for convenience as it simplifies calculation. However, other benchmarks like those mentioned above could have been chosen as well. With $\overline{\alpha}$ defined as the average return minus benchmark ($MAR = 0$), the Sortino ratio can be stated as

$$Sortino = \frac{\overline{\alpha}}{DD}. \tag{13.18}$$

The results obtained from the trading rules using (13.18) are based on daily data. To annualize the results, (13.18) is scaled by $\sqrt{m}$, i.e.

$$Sortino(p.a.) = \frac{\overline{\alpha}}{DD} \times \sqrt{m}. \tag{13.19}$$

All Sortino ratios reported in the subsequent discussion are annualized with $m = 253$ for the DAX and $m = 247$ for the Hang Seng, respectively. Now that fitness calculation of GP trading rules has been discussed it is time to elaborate on the choice of datasets on which the trading rules are generated and tested.

13.2.5 Choice of In- and Out-of-Sample Periods

As seen in Allen and Karjalainen (1999), Neely (2003b) and others, sample periods for GP-optimized trading rules are typically divided into an in-sample training and consecutive out-of-sample application period. In contrast to other contributions[59], the author chose a rather recent sample (1997-2007) for either

[59] See for example Allen and Karjalainen (1999) who use data from 1929-1995.

index. Consequently, the extremely long periods in Allen and Karjalainen (1999) are not feasible[60]. As training sample, 3-, and 5-years have been chosen which is roughly in line with existing literature. The idea is that longer in-sample periods provide more training input to the GP algorithm so that it can potentially pick up the distinctive features of a time series more easily. Shorter training periods (less than 3 years) do not seem sensible and are not used in any of the existing papers the author is aware of. The author conducted some experiments with 10-year training periods but due to the limits of the overall data sample, this scenario could only be tested with a single one-year out-of-sample period. Unfortunately, the results were poor despite the possibility for GP to train with a set of data that includes a full economic cycle. Consequently, the author did not further investigate this issue and results will not be covered further in the subsequent analysis.

Out-of-sample periods in the thesis range from 1- to 3-years. The rationale for not using longer periods is that the author believes that nowadays markets are so fast-paced and in constant change that any long-term GP approach is likely ill-fated. This view is supported by Ammann and Zenkner (2003) who use a rather recent data sample of the S&P 500 and find that even the best rules (which at best match the performance of buy-and-hold) implode after less than 2 years time. To summarize, the following periods have been analyzed:

Training periods in-sample : out-of-sample

- 3:1
- 3:2
- 3:3
- 5:1
- 5:2
- 5:3

for either index. For example, 5:2 reads "5-years in-sample training followed by 2-years out-of-sample data" and so on.

[60] The DAX was nonexistent until 1987.

A rolling time window approach extensively used throughout the literature[61] has been adopted for the upcoming analysis. For a 3-year training period, the first 3 years (1997-1998-1999) are taken as breeding ground for GP-optimized trading rules. The best rule found during a run on the training sample is then applied out-of-sample, i.e. the year (2000) data for the 1-year out-of-sample scenario. The time window then rolls to the right on an imaginary time scale axis. The next training sample is comprised of the years (1998-1999-2000), out-of-sample is (2001) and so on. The same applies to the 5-year training samples and the 2-and 3-years out-of-sample scenarios.

The rationale behind the use of separate in-sample training/learning and out-of-sample application data is obvious. As the most promising rule has evolved from training data and is subsequently applied out-of-sample, the final result is not based on ex-post data snooping. So in a certain sense, GP-optimized trading rules are ex-ante optimal.

As a final remark, some contributions such as as Allen and Karjalainen (1999) and Neely (2003b) make use of so-called validation periods set in between the training and out-of-sample period. Promising trading rules are first tested during the validation period and the best rule is then allowed to proceed to out-of-sample testing. Therefore, a further level of selection is introduced so that, in theory, the problem of overfitting is alleviated. While the concept is intuitively appealing, Navet and Chen (2007) point out that "the usefulness of validation,..., is still an open question" in their recent survey article. They argue that validation may be useful in stable markets[62] but also point out that it may be potentially harmful when markets are moving fast into different directions. Rules learned during the training sample would be outdated when they finally reach the out-of-sample phase thus resulting in poor performance.

Having just argued that out-of-sample periods have been deliberately chosen to be quite short (1-3 years) in order to adapt to today's fast-paced markets,

[61] See Allen and Karjalainen (1999), Ammann and Zenkner (2003) and Neely (2003b).
[62] Meaning markets where training, validation and out-of-sample set roughly feature the same pattern.

the author of the thesis chose not to include a validation period[63]. Last but not least, Chen and Kuo (2003a) even question the usefulness of validation periods.

13.2.6 Genetic Programming Setup

First of all, the issue of parameter choice for GP is nicely commented in Chen et al. (2008): "In particular, GP is notorious for its large number of user-supplied parameters, and the current research is not enough to allow us to inquire whether these parameters may impact the performance of GP."

No less than 62 parameters have to be set in GPLAB prior to a run with some of them being paramount such as how to initiate the very first generation (ramped half-and-half), which sampling method to use (fitness-proportional selection) etc. while others are less important and are only concerned with administrative overhead. In order to avoid excessive discussion of parameter settings, only the most important ones will be briefly addressed. Further information on the complete parameter set used is available from the author upon request.

The initial generation is created using the ramped half-and-half method[64] as suggested by Koza (1992) to ensure a sufficient variety of starting individuals to choose from. Selection of individuals is based on fitness-proportional selection[65]. The two main operators driving evolution are crossover and mutation. Both are used with automatic probability adjustment[66]. The third operator used is reproduction (copy & paste of individuals between two generations) which is always fixed at 10% probability throughout all GP runs.

While the overall goal of the thesis is to test for market efficiency making as good use of GP methodology as possible, some minor confessions are necessary

[63] As a side effect, inclusion of a validation period would have resulted in higher program code complexity and even more CPU time.

[64] See 9.1.4.

[65] See chapter 9.3. GPLAB offers tournament selection as well, however this did not improve upon the results during some casual testing.

[66] This basically means that after creation of the initial generation, mutation and crossover probabilities are chosen at random. The algorithm keeps track of the origin of the individuals through several subsequent generations and adapts operator probabilities based on the fitness of the individuals obtained so far. If the fittest individuals so far originated from crossover then crossover probability is increased in subsequent generations, mutation probability is decreased accordingly and vice versa. More details can be found in Davis (1989).

to make the approach feasible. One of these confessions is that total depth of trees (and thus trading rule complexity) is limited to 7 levels rather than allowing GP to freely evolve rules up to very high complexity. This has two reasons: First, the aim is to find trading rules that are as easily interpretable as possible (despite the limits set, trees at times still tend to get quite complex and thus difficult to interpret). Second, from a practical point of view it all boils down to CPU time. In order to get acceptable performance, tree complexity has to be limited. Furthermore, is is noteworthy that more complex trees do not necessarily yield better solutions. The results might even be worse due to the risk of overfitting[67]. Chen et al. (2008) find that more complex rules are not correlated with higher profits. Even more interesting, node complexity of the successful rules is very often less than 10 in all stock markets used in the study.

In this context, another parameter comes into play. Based on Silva and Almeida (2003), GPLAB offers to set a dynamic depth limit. This basically means that a dynamic slack depth limit is maintained to keep solutions as simple as possible unless a new solution found is more complex *and* superior to the solutions found before[68]. In this case, the depth limit is slackened to accomodate the more complex but better rule. However, there is still an absolute depth limit in place which eventually overrides the dynamic deepening of trees. Though this technique has not been used in any financial application the author is aware of, it has proved to contain complexity/bloat quite effectively in non-financial applications[69].

Another parameter deals with which individuals should enter a new generation. For all runs, the parameter was set to "replace" which basically means that once all individuals for a new generation have been created, all of them are used to completely replace the former generation regardless of their fitness[70]. This is

[67] See Koza (1992) for details.
[68] The dynamic depth limit is set automatically. For example, assuming that the best solution from the initial generation consists of 4 levels, the dynamic level could be set to 4. Consequently, all individuals of the next generation that do not comply with this setting have to be fitter than the existing solutions that set the standard, otherwise they will be rejected. After the second generation, a new dynamic level may be set and so on.
[69] See Silva and Almeida (2003).
[70] Newer individuals need not be fitter than older ones by definition. GPLAB offers elitism as

also called generational mode.

Next is the function set used to link the terminals consisting of closing prices, moving averages, maxima/minima, and lagged prices. The choice of appropriate function sets for GP has been rarely addressed in the literature. Wang and Soule (2004) investigate the performance of different function sets for different problems but the results are of little use for the application at hand. Most notably, they point out that a too large function set may unnecessarily increase the search space making it difficult for GP to find good solutions.

Therefore, the choice of a suitable function set is just based on intuition and a basic idea of what the rules might look like[71]. The functions used in all runs are:

- $+, -, \times, \div$
- $<, >$
- ln, *sqrt*
- and, or.

The operators "and" and "or" are of type boolean[72] with "and" evaluating as true if the input arguments are both nonzero and "or" evaluating as true unless the two input arguments are both zero[73]. More functions/operators could have been used as well but some considerations on algorithm efficiency led to the rejection of a broader function set. As already pointed out, the availability of too many functions to choose from may result in a dramatically increased search space rendering GP unable to find acceptable solutions at all[74]. Therefore, functions such as *exp*, *sin*, *cos* have been deliberately excluded from the function set. Navet and Chen (2007) point out the lack of guidelines for choosing a suitable function set so the decision of omitting *exp*, *sin*, *cos* is rather arbitrary. Apart from theoretical aspects, some thoughts on the nature of trading rules

an alternative. In this case, the best 50% of the old generation and the best 50% of the newly created individuals enter the new generation.

[71] This may sound like a contradiction as one of GP's most appealing features is its ability to come up with innovative and in some way "far-fetched" solutions a human mind would have never thought of. However, for practical reasons some sort of limited function set has to be used.

[72] Boolean means that they either evaluate as *true=1* or *false=0*.

[73] This will become clearer later when dealing with the structure of trading rules.

[74] See Koza (1992), Chen et al. (2002), Wang and Soule (2004), Navet and Chen (2007).

support the choice of a limited function set. For example, operators like *not*, $\leq$ and $\geq$ have been deliberately omitted as in the first case it might be easier for the algorithm to find positive rather than negated trading rules. Concerning the second and third case, it is unlikely that strict equality will arise during a run[75]. In contrast, the non-strict cases $<$ and $>$ are contained in the standard function set and may be considered important from an intuitive point of view. With their inclusion, simple and well-known popular strategies like trading range breaks and trend-following rules are possible with GP. These basic rules may be valuable building blocks for more elaborate rules depending on the "creativity" of GP. Therefore, it seems sensible to include these functions in the function set.

Last but not least the choice of number of individuals and generations for a run has to be made. The existing financial literature rarely elaborates on this. Neely et al. (1997) and Wang (2000) use 100 generations with 100 individuals[76]. However, Koza (1992) already pointed out the inherent risk of overfitting when using too many generations and individuals[77]. This was experimentally confirmed by Chen and Kuo (2003a) using synthetic time series[78].

Therefore, and for the sake of saving CPU time, the author stuck with the 25/50 approach (meaning 25 generations with 50 individuals each) advocated by Koza (1992) because it has proved to deliver satisfying results for a number of different applications using acceptable CPU time[79]. Furthermore, from an empirical finance point of view, the choice of "just" 25 generations is in-line with and Ammann and Zenkner (2003) and Drezewski and Sepielak (2008) who report that using considerably more generations only marginally improves fitness.

[75] As far as input matrices are concerned, it would be a rare coincidence that two input values are the same. By using basic arithmetic operators GP might be able to to come up with an equality during the course of a run, however with six decimal places this is highly unlikely.

[76] As a sidenote, Neely et al. (1997) report that it took several weeks to compute the results on a workstation.

[77] Which interestingly leads again to the question what "too many" means. As already pointed out, there are no theoretical results concerning optimal parameter choice in GP, at least in GP applications to financial time series.

[78] Interestingly, they also find that underfitting more likely occurs than overfitting in their experiments. However, they do not investigate the delicate balance between too few (=underfitting) generations/indivduals and too many (=overfitting).

[79] Fortunately, GP algorithms theoretically scale almost linearly and not exponentially (it might be more than linear if more complex trees than before are created within the extra generations).

This was confirmed for the application at hand by some experiments with more individuals and generations conducted by the author which did not improve upon the results[80] and in some cases resulted in significant overfitting. Consequently, and to save CPU time, the 25/50-approach was adopted. All results have been computed on an Intel Core 2 Duo 2.2 Ghz, 4GB RAM running Matlab R2007b on Mac OS-X 10.5.6.

Now that data inputs, fitness calculation and GP setup have been discussed it is time to take a look at the results obtained from the GP algorithm.

13.3 Genetic Programming Market Efficiency Tests

13.3.1 Testing the DAX

13.3.1.1 Introductory Remarks

The results obtained for the DAX are presented in Tables 13.3a - 13.8b. Each 13.Xa table is accompanied by a respective 13.Xb table containing some additional figures to help analyzing the results. The 13.Xa tables feature excess return, excess Sortino ratio, number of trades, number of buy- and sell-days, volatility during buy- and sell-days[81] and mean return thereof for three different levels of transaction costs.

In spirit of Allen and Karjalainen (1999) and Pereira (2002), statistical tests can be used to extend the analysis. The tables indicate the difference between mean daily GP-returns during buy- and sell-days ($\bar{r}_b - \bar{r}_s$) and the difference between GP-buy-days and "buy-and-hold-buy-days"[82] ($\bar{r}_b - \bar{r}_m$), respectively. It is straightforward to check the results for significance using a t-test. The first test statistics for GP-buy- and sell-days is defined as:

[80] Convergence, meaning that fitness does not improve anymore throughout further generations, was usually achieved after 15-20 generations during runs.

[81] Allen and Karjalainen (1999) find that volatility is lower on GP buy-days than sell-days which inspired the author to look at volatility as well.

[82] Which is of course always in-the-market.

$$t_{buy-sell} = \frac{\bar{r}_{buy} - \bar{r}_{sell}}{\sigma_{pool}\sqrt{\frac{1}{N_{buy}} + \frac{1}{N_{sell}}}} \tag{13.20}$$

where σ_{pool} denotes the pooled variance

$$\sigma_{pool} = \sqrt{\frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}} \tag{13.21}$$

with $n_{1,2}$ and $s_{1,2}$ denoting the respective sample size and standard deviation. Hypothesis testing works as follows:

$$H_0 : \bar{r}_{buy} - \bar{r}_{sell} \leq 0$$

$$H_1 : \bar{r}_{buy} - \bar{r}_{sell} > 0.$$

In a similar fashion, testing the difference between GP-buy and "buy-and-hold-buy"-days is done via

$$t_{buy} = \frac{\bar{r}_{buy} - \bar{r}_{m}}{\sigma_{pool}\sqrt{\frac{1}{N_{buy}} + \frac{1}{N}}} \tag{13.22}$$

with σ_{pool} defined as in (13.21) and

$$H_0 : \bar{r}_{buy} - \bar{r}_{m} \leq 0$$

$$H_1 : \bar{r}_{buy} - \bar{r}_{m} > 0.$$

Tables 13.Xb contain the returns from the GP-rules and buy-and-hold plus the difference between the two on an annualized basis. The same applies to the Sortino ratios for GP and buy-and-hold.

As a final remark before discussing the tables, it is tempting to relate the results to the existing literature presented in the second chapter of the thesis. However, it is quite difficult to draw comparisons since these studies are often based on similar but not exactly the same methodology (GA rather than GP or GP hybridized with fuzzy systems or neural nets) and were obtained from different stock markets or even asset classes such as futures and FOREX[83]. As another hurdle, the studies make use of differing total data samples, differing data division schemes, differing trading rule inputs and differing parameter settings, with the latter being particularly unhelpful for comparing results[84]. Therefore, only very few points in terms of relating the results obtained to existing studies will be made in the upcoming discussion and, due to the reasons just mentioned, should be taken with care.

13.3.1.2 Test Results

As a first impression, GP-generated trading rules generally yield negative results in terms of excess return and, even more important, excess Sortino ratio throughout most of the scenarios in Tables 13.3a - 13.8a. These key figures may imply that GP fails at beating a buy-and-hold strategy. However, further analysis reveals some cases of what may be some subtle inefficiencies in a highly mature and liquid market.

Apart from two cases of buy-and-hold, two successful GP trading rules are reported in Table 13.3a, Panel A, namely the 99-01/02 scenario and the 04-06/07 scenario. The first one is a special case since the out-of-sample year 2002 was marked by huge losses in the wake of Sept. 11th. The rule avoids a stock market investment most of the time which results in greatly reduced losses vs. buy-and-hold. The successful 04-06/07 rule surpasses buy-and-hold on a risk-adjusted basis as well. Another point worth mentioning is that the 02-04/05 and the 04-06/07 apparently have some forecasting power in terms of market direction, though the evidence is weak. The statistical significance for the 04-06/07 rule may explain the superior excess return and Sortino ratio. The algorithm apparently learned to distinguish between good and bad days in the index and switched out-of-the-market when returns were negative. The rule is long most

[83] Only Chen et al. (2008) briefly address the DAX, fortunately with “plain-vanilla GP”, whereas Setzkorn et al. (1996) do not elaborate on their results. The Hang Seng has not been covered at all by the existing literature to the best of the author’s knowledge.

[84] Even if all other ingridients were equal, a slight difference in parameters alone or even a single parameter may result in considerable changes in GP performance. Navet and Chen (2007) point out that the impact of changes in parameter settings on GP performance is not well understood.

of the time earning positive returns which by design results in a superior Sortino ratio. However, one has to bear in mind that the trading rule was generated under unrealistically low transaction costs of 0.1%. This may have led GP to trade too frequently during in-sample training resulting in a high in-sample fitness. Even though the rule outperforms buy-and-hold during out-of-sample testing (interestingly with just two trades), the respective rule perhaps would not have emerged in-sample under a more realistic (meaning higher) transaction costs. As a consequence, the 0.1% transaction cost scenarios are unrealistic and mainly serve for robustness checks rather than drawing inference about market efficiency. Therefore, they will not be addressed in-depth in further discussion. This applies to all scenarios.

The results under a more realistic transaction cost of 0.25% comply with the EMH as the best rules found are either buy-and-hold or negative in terms of excess return and excess Sortino ratio. During the later out-of-sample years, GP lags well behind the benchmark which is not too surprising given the sustained upward trend in the index throughout this period[85] making it very hard to beat the benchmark.

Panel C in Table 13.3a yields some interesting results as well. Excess returns and Sortino ratios are positive in three out of eight scenarios with one being the special post Sept. 11th case. Interestingly, the best rule found is the same as in the 0.1% transaction cost case. Despite different transaction costs, the algorithm comes up with the same trading rule which apparently shows that there seems to be no better solution from a GP point of view. Another point worth mentioning in this context is that transaction costs do not seem to have an impact on trading frequency as the frequency is basically the same regardless of transaction costs. This will be elaborated upon later in more detail. The 97-99/00 and the 04-06/07 case beat buy-and hold though the excess Sortino ratio is only marginally positive. In addition, the two GP-rules are quite mundane since they stay out-of-the-market for just a couple of days so they may be termed "smart buy-and-hold". Apart from excess returns/Sortino ratios, it is

[85] See Figure 13.1.

noteworthy that the volatility during buy-days tends to be slightly higher than during sell-days. This applies to all levels of transaction costs. Furthermore, $(\bar{r}_b - \bar{r}_m)$ is never significant.

Stretching the out-of-sample period to two years as reported in Table 13.4a generally does not seem to change the picture. For $c = 0.25$, results are negative across the board except for a single buy-and-hold case. The first rule for $c = 0.5$ yields a marginal excess return and Sortino ratio. The rule once more is almost-buy-and-hold ("smart buy-and-hold"). The 98-00/01-02 rule underperforms the benchmark in contrast to Chen et al. (2008) who report a statistically significant GP outperformance for the 2001-2002 DAX out-of-sample period using their setup of 5-years training plus 5-years validation period followed by 2-years out-of-sample testing for $c = 0.5$. The post-Sept. 11th scenario (99-01/02-03) is more striking. The GP rule for the aforementioned period yields a considerable outperformance mainly due to prolonged out-of-the-market periods. Nevertheless the rule just earns an almost zero return in absolute terms (0.00567 p.a., see Table 13.4b). This is mainly the same story told for the 1-year out-of-sample results (Table 13.3a, 13.3b) where the same rule yields a zero return (0.002771 p.a.) as well. However, the rule looks better in terms of excess return and Sortino ratio because buy-and-hold sustained tremendous losses during the first quarters after Sept. 11th whereas the market slightly recovered during the subsequent quarters which are covered in the 2-years out-of-sample scenario. Due to this slight recovery, the gap between GP and buy-and-hold narrows. The 00-02/03-04 scenario clearly misses the benchmark which is in line with Chen et al. (2008) for the same period.

As seen before, $(\bar{r}_b - \bar{r}_m)$ is always insignificant as was the case in the 1-year out-of-sample scenario.

Another case of the post-Sept. 11th market condition is depicted in Table 13.5a Panel C (3-years out-of-sample). Absolute returns are slightly negative this time (-0.010501 p.a., see Table 13.5b) and the gap between GP and buy-and-hold narrows once more. Volatility between buy- and sell-days is roughly equal.

The other rules in Table 13.5a are not particularly interesting due to either poor performance or unrealistic transaction costs.

The results obtained from extending the training sample from 3- to 5-years are next on the agenda. Table 13.6a once more highlights the post-Sept. 11th market conditions. The trading rule derived under $c = 0.5$ even manages to outperform the rule derived under $c = 0.25$[86]. The latter excessively jumps in- and out-of-the-market[87] and volatility during buy-days is considerably higher. In addition, the rule is insignificant in terms of market direction forecasting power[88].

The $c = 0.5$ rule just executes a single trade and maintains a prolonged out-of-the-market position. In contrast to the $c = 0.25$ rule, volatility during buy-days is considerably lower than during sell-days. Interestingly, the 98-02/03 rule in Panel B fails in terms of excess return but yields a better Sortino ratio than the benchmark investment[89].

With a two-year out-of-sample period (Table 13.7a, 13.7b), two rules with positive excess return and Sortino ratio emerge. The $c = 0.25$ rule gets into calmer water and just executes 6 more trades in the second year after 15 trades during the first out-of-sample year (Table 13.6a) resulting in 21 trades in total. As seen before, volatility during buy-days is still higher than during sell-days.

The $c = 0.5$ rule just adds another trade during the second out-of-sample year and shares the results observed during 1-year out-of-sample, namely lower volatility during buy-days. The other rules reported in Table 13.7a perform poorly across the board.

[86] However, both rules still yield negative absolute returns (-0.293879 p.a./-0.043641 p.a., see Table 13.6b.

[87] 15 trades within a single year is quite a lot compared to the rules discussed so far.

[88] This is unfortunate as the rule is somewhat intriguing since it manages to outperform buy-and-hold not by simply staying out-of-the-market but by jumping right into the market 15 times even in such a poor market condition as seen in 2002, yet it manages to outperform buy-and-hold. However, the higher volatility coupled with the insignificant market timing is somewhat disappointing.

[89] This is the only case where excess returns are negative but risk-adjusted returns positive.

For the 5:3 scenario in Table 13.8a, the 97-01/02-04 rule for $c = 0.25$ and $c = 0.5$ stick out. Volatility during in-days is lower for the latter rule. As already seen before, the $c = 0.25$ rule trades quite often (though trading takes place only during the first two out-of-sample years, see Table 13.7a) and the $c = 0.5$ rule also does not enter into any new trades after the second out-of-sample year (see Table 13.7a). Consequently, excess returns and Sortino ratios, while still in positive territory, melt down. The $c = 0.25$ rule stays in-the-market during the third year which result in the same excess return but lower Sortino ratio compared to the 2-year out-of-sample case wheras the $c = 0.5$ strategy continues to stay out-of-the-market during the third out-of-sample year thus suffering both in terms of excess return and Sortino ratio due to a sustained upward trend in the benchmark[90].

At this stage, some comments on the impact of transaction costs on trading frequency are in order. As already seen in the scenarios using 3-years training, GP generally seems to be unaffacted by transaction costs which is somewhat counterintuitive since one would expect trading frequency to decrease when transaction costs increase. However, this behaviour can be observed in the 5-year training results. Though the pattern is weak during the 1-year out-of-sample case, it is more discernible in the 2- and 3-year out-of-sample results. It apparently takes some time until the effect emerges. In addition, GP does not seem to care much about whether transaction costs are $c = 0.1$ or $c = 0.25$, but $c = 0.5$ seems to change the picture resulting in lower trading frequencies. Trading frequencies are generally speaking quite low which is in line with Navet and Chen (2007) and Chen et al. (2008)[91].

Summing up the most important results just presented, the following points can be made for the DAX:

- GP-generated trading rules fail at consistently beating buy-and-hold on a risk-adjusted basis thus indicating market efficiency...

[90] The timing of a GP trading rule (0=money market, 1=stock market) can be easily visualized on a horizontal time scale to reveal which position was taken at a particular day/month/year. However, the author felt that providing this chart for every single trading rule would not add much information and thus refrained from including it in the discussion.

[91] They report trading frequencies of 1-9 trades for two years out-of-sample.

- but at least 3 rules outperform buy-and-hold on a risk-adjusted basis in the wake of Sept. 11th
- 2 more rules marginally beat buy-and-hold by staying in-the-market except for a few days
- yet GP rules have no statistically significant forecasting power.

The last item is particularly important. The ability of GP to outperform the benchmark seems to be based on the ability to switch out-of-the market to avoid losses rather than picking the right in-days. However, it apparently does so on a level that is below statistical significance.

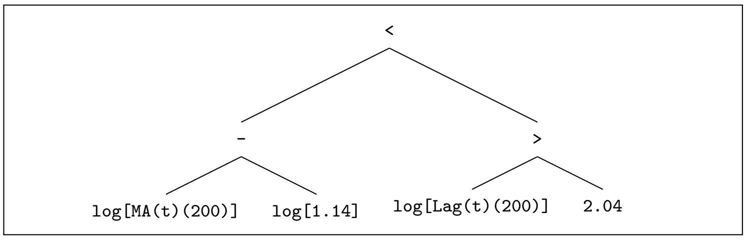
The successful rules found for the DAX are compiled in Table 13.9. In total, five rules yield positive excess returns and Sortino ratios whereas one rule yielded a negative excess return but positive Sortino ratio which is why it has been included as well in 13.9 for further analysis. At least three out of six rules[92] are affected by the post-Sept. 11th market turmoil where GP proved useful by finding ex-ante "near-optimal" rules that beat the market.

Leaving aside the post-Sept. 11th market conditions, the 97-99/...c05 rule yields slightly better risk-adjusted returns during the year 2000. However, the outperformance melts down when stretching the out-of-sample period one year further. The 04-06/...c05 rule is the "smart-buy-and-hold"-rule that is almost always long except for a couple of days and yields only marginal excess returns. As a general impression, excess Sortino ratios seem to decline over the years for the rules indicating that they lose power as time progresses which is in line with the findings in Ammann and Zenkner (2003).

[92] The 98-02/...c025 rule is debatable.

13.3.1.3 Structure of Trading Rules

In their survey paper, Chen and Kuo (2003b) focus entirely on the structure of trading rules found by GP in some of the studies discussed in the second chapter of the thesis. However, the usefulness of these results for the study at hand is limited due to the use of different markets, data samples etc. Nevertheless, the basic idea of taking a closer look at the succesful trading rules seems appealing. Three out of six successful rules from table 13.9 are depicted in Figure 13.3a-c. The rules that do not show up in Figure 13.3 are quite complex and have no straightforward interpretation. Figure 13.3a depicts the trading rule bred on the 04-06c05 training sample. As already pointed, the rule is of the "smart buy-and-hold" type meaning that it mostly stays in-the-market during out-of-sample testing except for a few days. The structure of the rule might not look intuitive at first, however the rule is quite easy as it will turn out later. The "mylog" operator is a protected function[93] and takes the log of its argument. If the argument is negative, the absolute value is taken as argument instead. Upon further inspection, it turns out that the tree in Figure 13.3a collapses into



which is a somewhat simpler representation. As `Lag(t)(200)` is in the range 1.10-1.15, `log[Lag(t)(200)]` will always be `< 2.04` so the right hand side sub-tree will always be zero (=false). The tree evaluates to true if `MA(t)(200)<1.14` so the rule basically boils down to "go long if `MA(t)(200)<1.14`, else stay out"[94]. Therefore, the rule has intuitive appeal in economic terms as it might be thought of as protection against an overheating market just like an electrical fuse that melts when too much current flows.

[93] See 9.1.2.

[94] As a reminder, the rules are based on *normalized* closing prices rather than the original price series.

Figure 13.3b depicts the trading rule obtained from the 99-01c05 training sample. The rule is rather self-explaining and takes an in-the-market position if the product of `Min(t)(100)` and `Lag(t)(250)` is smaller than `Min(t)(200)`. For the special case of a 1-year out-of-sample period, the strategy is a seasonal rule that enters the market after 200 trading days (rougly end of october) since `Lag(t)(250)` is undefined=zero in computer terms prior to the 250th day of trading resulting in a zero subtree on the left. `Min(t)(200)` is nonzero after 200 days of trading so the rule evaluates as true and goes long after 200 days of trading. Last but not least, Figure 13.3c stems from the 97-01c05 scenario. The top of the tree features the boolean operator "and" that evaluates to true only if both of its arguments are $\neq 0$. The left-hand-side subtree almost always meets this requirement[95]. Consequently, the right-hand side subtree tips the scales. Leaving aside the "mylog" operator, the subtree returns 1=true if and only if `Min(t)(100)` is greater than 0.99. Therefore, the rule goes long if the market has shown some signs of robustness over the last 100 trading days, else it stays out-of-the-market.

It is noteworthy that the successful rules just discussed tend to rely on long-term indicators (100, 200 and 250 days) rather than short-term indicators. This also applies to the more complex rules not depicted in Figure 13.3. The presence of long-term indicators implies that GP picks up long-term trends in the data rather than reacting to short-term noise. As a by-product, the presence of long-term indicators in GP trading rules might also explain the overall low trading frequencies that have been observed so far. This issue will be elaborated upon in the upcoming discussion of the results for the Hang Seng.

[95] It is highly unlikely that `Min(t)(10)==MA(t)(50)`, especially when using six digit decimal places.

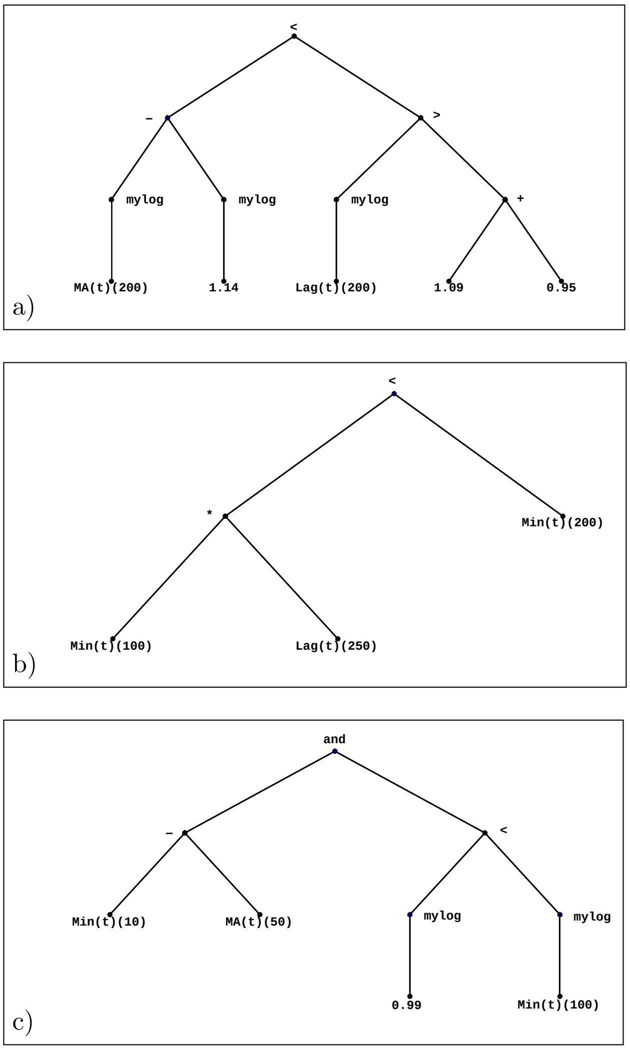


Figure 13.3: Tree structure of successful DAX trading rules.

13.3.1.4 Long Term Genetic Programming Performance

Another issue is investigated in Figure 13.4: What would happen if an investor followed a GP revolving strategy over the whole data sample using a 1-year out-of-sample time frame with the GP trading rule updated every year[96]? This question is particularly important for determining whether the market has been generally efficient or not. Adding up log-returns yields the so-called equity curve which gives a clear picture as to whether and when GP was superior (or inferior)

[96] I.e. using the rule learnt from the 97-98-99 sample in 2000 and then for 2001 the rule learnt during 98-99-00 and so on..

to buy-and-hold in the long run[97]. The 3:1c025 revolving GP strategy performs poorly whereas the 3:1c05, 5:1c025 and 5:1c05 strategies seem to have an edge at first since they manage to avoid some losses during the post-Sept. 11th market depression. However, buy-and-hold eventually catches up and surpasses GP returns due to a strong and sustained upward trend in the DAX which favors buy-and-hold as benchmark. GP jumps into the market as well in the 3:1c05 scenario but too late to catch up. GP entirely misses the late buy-and-hold trend in the 5:1c025 and 5:1c05 scenarios. Summary statistics have been calculated as well, however the results are not shown here since they do not add much to the story told by Figure 13.4.

The only notable exception is the 5:1c05 scenario. Interestingly, GP manages to stay well above buy-and-hold for a long time but is eventually overtaken by the benchmark in 2006. Summary statistics in table Table 13.2 show that GP finishes below buy-and-hold in terms of total return (0.259 vs 0.436) but manages to beat the benchmark in risk-adjusted terms (0.0344 vs. 0.0265), albeit marginally. In addition, the standard deviation for GP is considerably lower mainly due to some prolonged out-of-the-market positions. Even though the overall findings just reported apparently comply with the EMH, it is noteworthy that in the last case just mentioned, GP matches (and marginally outperforms on a risk-adjusted level) buy-and-hold. Even though GP long term performance might not be impressive, its ability to at least match the benchmark in the long run (2002-2007) proves that GP is a suitable and powerful technique for financial knowledge discovery justifying the effort taken in the study at hand.

Last but not least, another perspective on return distributions is provided in Figure 13.5. While it does not provide a lot of additional information, it is still noteworthy that the 3:1 scenarios result in a spiked return distribution around the mean, whereas the 5:1 distributions feature a more jagged shape.

[97] A straight almost horizontal line indicates a prolonged out-of-the-market position earning the money market rate.

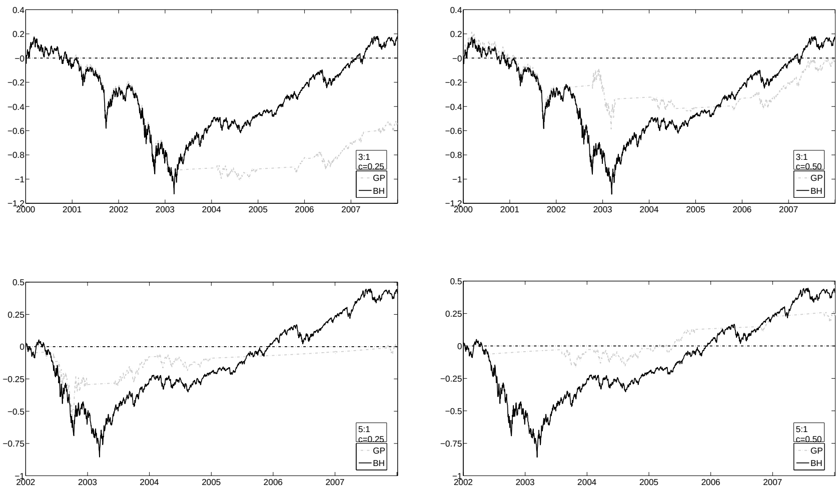


Figure 13.4: *Equity curves for 3:1 and 5:1 revolving Genetic Programming strategies for the DAX for c=0.25 and c=0.5.*

	GP	*BH*
Sample	2002-2007	2002-2007
Mean	0.000170	0.000286
Median	0.000101	0.001023
Minimum	-0.035162	-0.063360
Maximum	0.031551	0.075527
Std.Dev.	0.006890	0.015116
Skewness	-0.351833	-0.023670
Ex.Kurtosis	5.172195	3.543065
Total Return	0.259623	0.436237
Sortino Ratio	0.034445	0.026597

Table 13.2: DAX 5:1 $c = 0.5$ revolving strategy results.

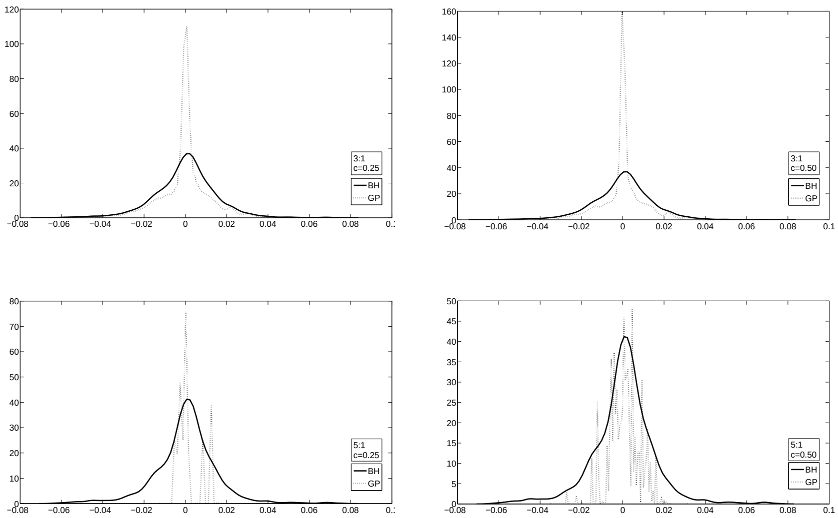


Figure 13.5: *Kernel smoothing density estimates for 3:1 and 5:1 DAX scenarios for c=0.25 and c=0.5.*

Sample	Excess	ΔSOR	$\#T$	N_b	N_s	σ_b	σ_s	$\bar{r}_b$	$\bar{r}_s$	$(\bar{r}_b - \bar{r}_s)$	$(\bar{r}_b - \bar{r}_m)$
Panel A: Transaction costs 0.1%											
97 − 99/00	0.000000	0.000000	1	253	0	0.015007	0.000000	−0.000190	0.000000	−0.000190	0.000000
98 − 00/01	0.000000	0.000000	1	251	0	0.018029	0.000000	−0.000789	0.000000	−0.000789	0.000000
99 − 01/02	0.593067	1.493110	1	53	199	0.029606	0.024000	−0.000247	−0.002850	0.002603	0.002056
00 − 02/03	−0.219194	−0.795403	0	0	252	0.000000	0.019312	0.000000	0.000970	−0.000970	−0.000970
01 − 03/04	−0.042673	−0.258568	1	86	170	0.010717	0.009498	0.000010	0.000333	−0.000322	−0.000214
02 − 04/05	−0.147004	−1.174332	1	23	233	0.008405	0.007550	0.002819	0.000714	0.002105*	0.001916
03 − 05/06	−0.031359	−0.198891	2	179	75	0.009888	0.009375	0.000862	0.000489	0.000373	0.000110
04 − 06/07	0.070303	0.391734	2	225	26	0.009961	0.007174	0.001141	−0.002619	0.003759*	0.000389
Panel B: Transaction costs 0.25%											
97 − 99/00	0.000000	0.000000	1	253	0	0.015007	0.000000	−0.000190	0.000000	−0.000190	0.000000
98 − 00/01	0.000000	0.000000	1	251	0	0.018029	0.000000	−0.000789	0.000000	−0.000789	0.000000
99 − 01/02	0.000000	0.000000	1	252	0	0.025241	0.000000	−0.002303	0.000000	−0.002303	0.000000
00 − 02/03	−0.322151	−1.011389	1	15	237	0.018851	0.019279	−0.006627	0.001451	−0.008078	−0.007598
01 − 03/04	−0.061958	−0.362580	5	205	51	0.010334	0.008002	0.000055	0.000904	−0.000849	−0.000169
02 − 04/05	−0.142851	−1.141710	1	57	199	0.007436	0.007709	0.001263	0.000800	0.000463	0.000360
03 − 05/06	−0.058467	−0.356218	1	205	49	0.009858	0.009212	0.000624	0.001288	−0.000664	−0.000128
04 − 06/07	−0.011519	−0.033871	3	153	98	0.009572	0.010083	0.001125	0.000167	0.000959	0.000374
Panel C: Transaction costs 0.5%											
97 − 99/00	0.032598	0.135917	1	249	4	0.015010	0.014447	−0.000065	−0.008010	0.007945	0.000126
98 − 00/01	−0.017135	−0.054463	1	246	5	0.018140	0.011675	−0.000878	0.003614	−0.004493	−0.000089
99 − 01/02	0.593067	1.496094	1	53	199	0.029606	0.024000	−0.000247	−0.002850	0.002603	0.002056
00 − 02/03	−0.320253	−0.973012	1	73	179	0.027079	0.015048	−0.001251	0.001876	−0.003127	−0.002221
01 − 03/04	−0.134965	−0.754849	3	160	96	0.010452	0.008871	−0.000408	0.001279	−0.001687	−0.000633
02 − 04/05	−0.141942	−1.123853	1	47	209	0.007786	0.007615	0.001532	0.000762	0.000770	0.000628
03 − 05/06	−0.058467	−0.354946	1	205	49	0.009858	0.009212	0.000624	0.001288	−0.000664	−0.000128
04 − 06/07	0.016128	0.090852	2	244	7	0.009824	0.006506	0.000875	−0.003570	0.004445	0.000124

Table 13.3a: 3-years training and 1-year out-of-sample DAX results. "Sample" denotes the training period used followed by the out-of-sample testing period (for example 97-99 implies that training data from 1997, 1998 and 1999 have been used to derive a trading rule which is then applied out-of-sample to data from 2000 and so on)."Excess" measures the fitness implied by a GP trading rule defined as excess return over a buy-and-hold strategy during the out-of-sample period, i.e. $(r_{gp} - r_{bh})$. ΔSOR indicates the excess Sortino ratio defined as $(SOR_{gp} - SOR_{bh})$ (both annualized) over the specified out-of-sample period. $\#T$ indicates the number of trades executed by a trading rule during out-of-sample testing with N_b and N_s denoting the number of buy-days (in-the-market) and sell-days (out-of-the-market), respectively. σ_b and σ_s indicate the standard deviation of returns during GP-in-market-days and GP-out-of-market-days, respectively. $\bar{r}_b$ and $\bar{r}_s$ denote the mean daily market return during GP-in-days and GP-out-days with $(\bar{r}_b - \bar{r}_s)$ as the difference between the two. $(\bar{r}_b - \bar{r}_m)$ measures the difference between mean daily returns during GP-in-days and buy-and-hold. (*) indicates significance for $\alpha = 0.05$.

Sample	r_{gp}	r_{bh}	Δr	SOR_{gp}	SOR_{bh}	ΔSOR
Panel A: Transaction costs 0.1%						
97 − 99/00	−0.050119	−0.050119	0.000000	−0.210933	−0.210933	0.000000
98 − 00/01	−0.199976	−0.199976	0.000000	−0.678159	−0.678159	0.000000
99 − 01/02	0.010771	−0.582296	0.593067	0.023130	−1.469980	1.493110
00 − 02/03	0.023326	0.242520	−0.219194	0.000000	0.795403	−0.795403
01 − 03/04	0.012767	0.055440	−0.042673	0.066978	0.325546	−0.258568
02 − 04/05	0.082280	0.229284	−0.147004	0.606102	1.780435	−1.174332
03 − 05/06	0.157632	0.188991	−0.031359	0.898631	1.097522	−0.198891
04 − 06/07	0.256838	0.186534	0.070303	1.541896	1.150162	0.391734
Panel B: Transaction costs 0.25%						
97 − 99/00	−0.053119	−0.053119	0.000000	−0.223249	−0.223249	0.000000
98 − 00/01	−0.202977	−0.202977	0.000000	−0.688333	−0.688333	0.000000
99 − 01/02	−0.585296	−0.585296	0.000000	−1.477554	−1.477554	0.000000
00 − 02/03	−0.082632	0.239520	−0.322151	−0.225974	0.785415	−1.011389
01 − 03/04	−0.009518	0.052440	−0.061958	−0.053374	0.309206	−0.362580
02 − 04/05	0.083433	0.226284	−0.142851	0.610348	1.752057	−1.141710
03 − 05/06	0.127524	0.185991	−0.058467	0.728466	1.084684	−0.356218
04 − 06/07	0.172015	0.183534	−0.011519	1.102879	1.136750	−0.033871
Panel C: Transaction costs 0.5%						
97 − 99/00	−0.025522	−0.058119	0.032598	−0.107743	−0.243660	0.135917
98 − 00/01	−0.225112	−0.207977	−0.017135	−0.759752	−0.705289	−0.054463
99 − 01/02	0.002771	−0.590297	0.593067	0.005918	−1.490176	1.496094
00 − 02/03	−0.085733	0.234520	−0.320253	−0.201179	0.771833	−0.973012
01 − 03/04	−0.087525	0.047440	−0.134965	−0.474071	0.280778	−0.754849
02 − 04/05	0.079342	0.221284	−0.141942	0.578799	1.702652	−1.123853
03 − 05/06	0.122524	0.180991	−0.058467	0.698896	1.053842	−0.354946
04 − 06/07	0.194662	0.178534	0.016128	1.200976	1.110124	0.090852

Table 13.3b: 3-years training and 1-year out-of-sample DAX results. "Sample" denotes the length of training and subsequent out-of-sample period. r_{gp} and r_{bh} denote the annualized out-of-sample returns for the GP trading rule and buy-and-hold, respectively. Δr is the difference between them. SOR_{gp} and SOR_{bh} indicate the repective annualized Sortino ratios for the GP trading rule and buy-and-hold, ΔSOR measures the difference between the two and is equal to ΔSOR in the preceding table.

Sample	Excess	ΔSOR	$\#T$	N_b	N_s	σ_b	σ_s	$\bar{r}_b$	$\bar{r}_s$	$(\bar{r}_b - \bar{r}_s)$	$(\bar{r}_b - \bar{r}_m)$
Panel A: Transaction costs 0.1%											
97 − 99/00 − 01	0.340002	0.650329	11	306	199	0.015268	0.018401	0.000184	−0.001634	0.001818	0.000716
98 − 00/01 − 02	0.000000	0.000000	1	504	0	0.021910	0.000000	−0.001541	0.000000	−0.001541	0.000000
99 − 01/02 − 03	0.593067	0.853595	1	306	199	0.021740	0.024000	0.000988	−0.002850	0.003838*	0.001512
00 − 02/03 − 04	−0.269127	−0.629786	0	0	509	0.000000	0.015296	0.000000	0.000619	−0.000619	−0.000619
01 − 03/04 − 05	−0.260818	−0.876914	1	86	427	0.010717	0.008421	0.000010	0.000693	−0.000683	−0.000569
02 − 04/05 − 06	−0.192654	−0.682303	2	177	334	0.009394	0.008360	0.001186	0.000659	0.000527	0.000345
03 − 05/06 − 07	−0.005972	0.000600	3	399	107	0.009583	0.010322	0.000947	0.000134	0.000813	0.000172
Panel B: Transaction costs 0.25%											
97 − 99/00 − 01	−0.055302	−0.096770	5	486	19	0.016679	0.013916	−0.000633	0.002043	−0.002676	−0.000101
98 − 00/01 − 02	0.000000	0.000000	1	504	0	0.021910	0.000000	−0.001541	0.000000	−0.001541	0.000000
99 − 01/02 − 03	−0.197471	−0.278721	6	484	21	0.022345	0.029248	−0.000908	0.008315	−0.009223	−0.000384
00 − 02/03 − 04	−0.372054	−0.708536	1	15	494	0.018851	0.015145	−0.006627	0.000840	−0.007467	−0.007247
01 − 03/04 − 05	−0.061958	−0.217305	5	462	51	0.008934	0.008002	0.000543	0.000904	−0.000361	−0.000036
02 − 04/05 − 06	−0.142851	−0.551709	1	312	199	0.009330	0.007709	0.000868	0.000800	0.000067	0.000026
03 − 05/06 − 07	−0.058467	−0.182371	1	457	49	0.009802	0.009212	0.000720	0.001288	−0.000567	−0.000055
Panel C: Transaction costs 0.5%											
97 − 99/00 − 01	0.005863	0.011110	5	498	7	0.016580	0.016690	−0.000450	−0.006399	0.005950	0.000082
98 − 00/01 − 02	−0.079929	−0.109888	2	498	6	0.021875	0.022677	−0.001702	0.011832	−0.013535	−0.000161
99 − 01/02 − 03	0.276050	0.386905	8	203	302	0.022228	0.023055	0.000227	−0.001030	0.001256	0.000751
00 − 02/03 − 04	−0.370150	−0.689953	1	73	436	0.027079	0.012296	−0.001251	0.000933	−0.002183	−0.001870
01 − 03/04 − 05	−0.134965	−0.455876	3	417	96	0.008835	0.008871	0.000418	0.001279	−0.000861	−0.000161
02 − 04/05 − 06	−0.167504	−0.596674	2	204	307	0.009392	0.008262	0.001203	0.000601	0.000602	0.000362
03 − 05/06 − 07	−0.058467	−0.182249	1	457	49	0.009802	0.009212	0.000720	0.001288	−0.000567	−0.000055

Table 13.4a: 3-years training and 2-years out-of-sample DAX results. "Sample" denotes the training period used followed by the out-of-sample testing period (for example 97-99 implies that training data from 1997, 1998 and 1999 have been used to derive a trading rule which is then applied out-of-sample to data from 2000 and so on)."Excess" measures the fitness implied by a GP trading rule defined as excess return over a buy-and-hold strategy during the out-of-sample period, i.e. $(r_{gp} - r_{bh})$. ΔSOR indicates the excess Sortino ratio defined as $(SOR_{gp} - SOR_{bh})$ (both annualized) over the specified out-of-sample period. $\#T$ indicates the number of trades executed by a trading rule during out-of-sample testing with N_b and N_s denoting the number of buy-days (in-the-market) and sell-days (out-of-the-market), respectively. σ_b and σ_s indicate the standard deviation of returns during GP-in-market-days and GP-out-of-market-days, respectively. $\bar{r}_b$ and $\bar{r}_s$ denote the mean daily market return during GP-in-days and GP-out-days with $(\bar{r}_b - \bar{r}_s)$ as the difference between the two. $(\bar{r}_b - \bar{r}_m)$ measures the difference between mean daily returns during GP-in-days and buy-and-hold. (*) indicates significance for $\alpha = 0.05$.

Sample	r_{gp}	r_{bh}	Δr	SOR_{gp}	SOR_{bh}	ΔSOR
Panel A: Transaction costs 0.1%						
97 − 99/00 − 01	0.034651	−0.135350	0.170001	0.147090	−0.503239	0.650329
98 − 00/01 − 02	−0.389383	−0.389383	0.000000	−1.105914	−1.105914	0.000000
99 − 01/02 − 03	0.163075	−0.133458	0.296534	0.480225	−0.373370	0.853595
00 − 02/03 − 04	0.022097	0.156661	−0.134564	0.000000	0.629786	−0.629786
01 − 03/04 − 05	0.017100	0.147509	−0.130409	0.089539	0.966453	−0.876914
02 − 04/05 − 06	0.117653	0.213980	−0.096327	0.721411	1.403714	−0.682303
03 − 05/06 − 07	0.192119	0.195105	−0.002986	1.165878	1.165278	0.000600
Panel B: Transaction costs 0.25%						
97 − 99/00 − 01	−0.164501	−0.136850	−0.027651	−0.605312	−0.508542	−0.096770
98 − 00/01 − 02	−0.390883	−0.390883	0.000000	−1.110175	−1.110175	0.000000
99 − 01/02 − 03	−0.233694	−0.134958	−0.098736	−0.656287	−0.377567	−0.278721
00 − 02/03 − 04	−0.030866	0.155161	−0.186027	−0.083580	0.624956	−0.708536
01 − 03/04 − 05	0.115030	0.146009	−0.030979	0.738535	0.955840	−0.217305
02 − 04/05 − 06	0.141054	0.212480	−0.071425	0.841502	1.393211	−0.551709
03 − 05/06 − 07	0.164371	0.193605	−0.029233	0.976627	1.158998	−0.182371
Panel C: Transaction costs 0.5%						
97 − 99/00 − 01	−0.136418	−0.139350	0.002931	−0.506225	−0.517335	0.011110
98 − 00/01 − 02	−0.433348	−0.393383	−0.039965	−1.227163	−1.117275	−0.109888
99 − 01/02 − 03	0.000567	−0.137458	0.138025	0.001614	−0.385292	0.386905
00 − 02/03 − 04	−0.032414	0.152661	−0.185075	−0.075315	0.614638	−0.689953
01 − 03/04 − 05	0.076027	0.143509	−0.067482	0.484154	0.940029	−0.455876
02 − 04/05 − 06	0.126228	0.209980	−0.083752	0.778330	1.375003	−0.596674
03 − 05/06 − 07	0.161871	0.191105	−0.029233	0.964165	1.146414	−0.182249

Table 13.4b: 3-years training and 2-years out-of-sample DAX results. “Sample” denotes the length of training and subsequent out-of-sample period. r_{gp} and r_{bh} denote the annualized out-of-sample returns for the GP trading rule and buy-and-hold, respectively. Δr is the difference between them. SOR_{gp} and SOR_{bh} indicate the repective annualized Sortino ratios for the GP trading rule and buy-and-hold, ΔSOR measures the difference between the two and is equal to ΔSOR in the preceding table.

Sample	Excess	ΔSOR	$\#T$	N_b	N_s	σ_b	σ_s	$\bar{r}_b$	$\bar{r}_s$	$(\bar{r}_b - \bar{r}_s)$	$(\bar{r}_b - \bar{r}_m)$
Panel A: Transaction costs 0.1%											
97 − 99/00 − 02	0.358853	0.379574	12	556	202	0.020430	0.018290	−0.000903	−0.001711	0.000808	0.000215
98 − 00/01 − 03	0.000000	0.000000	1	757	0	0.021246	0.000000	−0.000609	0.000000	−0.000609	0.000000
99 − 01/02 − 04	0.526382	0.547233	1	357	405	0.020456	0.018317	0.000811	−0.001194	0.002006	0.001066
00 − 02/03 − 05	−0.487261	−0.837831	0	0	766	0.000000	0.013225	0.000000	0.000724	−0.000724	−0.000724
01 − 03/04 − 06	−0.430754	−0.922701	1	86	682	0.010717	0.008919	0.000010	0.000726	−0.000715	−0.000635
02 − 04/05 − 07	−0.257972	−0.594507	4	293	470	0.009642	0.008716	0.001124	0.000642	0.000483	0.000297
Panel B: Transaction costs 0.25%											
97 − 99/00 − 02	−0.055302	−0.050500	5	739	19	0.020004	0.013916	−0.001199	0.002043	−0.003242	−0.000081
98 − 00/01 − 03	−0.041324	−0.040569	2	755	2	0.021244	0.016314	−0.000659	0.018246	−0.018906	−0.000050
99 − 01/02 − 04	−0.268285	−0.249279	6	592	170	0.020782	0.013177	−0.000763	0.001515	−0.002278	−0.000508
00 − 02/03 − 05	−0.590200	−0.869220	1	15	751	0.018851	0.013064	−0.006627	0.000871	−0.007498	−0.007352
01 − 03/04 − 06	−0.061958	−0.142627	5	717	51	0.009213	0.008002	0.000627	0.000904	−0.000277	−0.000018
02 − 04/05 − 07	−0.163251	−0.411723	4	559	204	0.009562	0.007625	0.000833	0.000811	0.000023	0.000006
Panel C: Transaction costs 0.5%											
97 − 99/00 − 02	−0.002551	−0.005856	8	748	10	0.019655	0.033489	−0.001045	−0.006599	0.005555	0.000073
98 − 00/01 − 03	−0.079929	−0.077673	2	751	6	0.021221	0.022677	−0.000709	0.011832	−0.012541	−0.000099
99 − 01/02 − 04	0.172612	0.184907	9	299	463	0.019481	0.019303	0.000034	−0.000441	0.000476	0.000289
00 − 02/03 − 05	−0.588337	−0.858267	1	73	693	0.027079	0.010799	−0.001251	0.000932	−0.002183	−0.001975
01 − 03/04 − 06	−0.254917	−0.545674	5	640	128	0.008986	0.009798	0.000421	0.001768	−0.001348	−0.000225
02 − 04/05 − 07	−0.167504	−0.396666	2	456	307	0.009597	0.008262	0.000980	0.000601	0.000379	0.000152

Table 13.5a: 3-years training and 3-years out-of-sample DAX results. "Sample" denotes the training period used followed by the out-of-sample testing period (for example 97-99 implies that training data from 1997, 1998 and 1999 have been used to derive a trading rule which is then applied out-of-sample to data from 2000 and so on)."Excess" measures the fitness implied by a GP trading rule defined as excess return over a buy-and-hold strategy during the out-of-sample period, i.e. $(r_{gp} - r_{bh})$. ΔSOR indicates the excess Sortino ratio defined as $(SOR_{gp} - SOR_{bh})$ (both annualized) over the specified out-of-sample period. $\#T$ indicates the number of trades executed by a trading rule during out-of-sample testing with N_b and N_s denoting the number of buy-days (in-the-market) and sell-days (out-of-the-market), respectively. σ_b and σ_s indicate the standard deviation of returns during GP-in-market-days and GP-out-of-market-days, respectively. $\bar{r}_b$ and $\bar{r}_s$ denote the mean daily market return during GP-in-days and GP-out-days with $(\bar{r}_b - \bar{r}_s)$ as the difference between the two. $(\bar{r}_b - \bar{r}_m)$ measures the difference between mean daily returns during GP-in-days and buy-and-hold.

Sample	r_{gp}	r_{bh}	Δr	SOR_{gp}	SOR_{bh}	ΔSOR
Panel A: Transaction costs 0.1%						
97 − 99/00 − 02	−0.163545	−0.283163	0.119618	−0.504142	−0.883716	0.379574
98 − 00/01 − 03	−0.154462	−0.154462	0.000000	−0.456285	−0.456285	0.000000
99 − 01/02 − 04	0.110089	−0.065371	0.175461	0.337941	−0.209292	0.547233
00 − 02/03 − 05	0.021880	0.184300	−0.162420	0.000000	0.837831	−0.837831
01 − 03/04 − 06	0.020980	0.164565	−0.143585	0.110069	1.032769	−0.922701
02 − 04/05 − 07	0.123735	0.209726	−0.085991	0.748416	1.342923	−0.594507
Panel B: Transaction costs 0.25%						
97 − 99/00 − 02	−0.302597	−0.284163	−0.018434	−0.937119	−0.886619	−0.050500
98 − 00/01 − 03	−0.169237	−0.155462	−0.013775	−0.499808	−0.459239	−0.040569
99 − 01/02 − 04	−0.155800	−0.066371	−0.089428	−0.462051	−0.212772	−0.249279
00 − 02/03 − 05	−0.013433	0.183300	−0.196733	−0.036256	0.832964	−0.869220
01 − 03/04 − 06	0.142912	0.163565	−0.020653	0.883675	1.026303	−0.142627
02 − 04/05 − 07	0.154309	0.208726	−0.054417	0.924667	1.336389	−0.411723
Panel C: Transaction costs 0.5%						
97 − 99/00 − 02	−0.286680	−0.285830	−0.000850	−0.897285	−0.891429	−0.005856
98 − 00/01 − 03	−0.183772	−0.157129	−0.026643	−0.542428	−0.464755	−0.077673
99 − 01/02 − 04	−0.010501	−0.068038	0.057537	−0.033201	−0.218109	0.184907
00 − 02/03 − 05	−0.014479	0.181634	−0.196112	−0.033532	0.824735	−0.858267
01 − 03/04 − 06	0.076926	0.161898	−0.084972	0.471228	1.016902	−0.545674
02 − 04/05 − 07	0.151225	0.207059	−0.055835	0.930645	1.327311	−0.396666

Table 13.5b: 3-years training and 3-years sout-of-sample DAX results. “Sample” denotes the length of training and subsequent out-of-sample period. r_{gp} and r_{bh} denote the annualized out-of-sample returns for the GP trading rule and buy-and-hold, respectively. Δr is the difference between them. SOR_{gp} and SOR_{bh} indicate the repective annualized Sortino ratios for the GP trading rule and buy-and-hold, ΔSOR measures the difference between the two and is equal to ΔSOR in the preceding table.

Sample	Excess	ΔSOR	$\#T$	N_b	N_s	σ_b	σ_s	$\bar{r}_b$	$\bar{r}_s$	$(\bar{r}_b - \bar{r}_s)$	$(\bar{r}_b - \bar{r}_m)$
Panel A: Transaction costs 0.1%											
97 − 01/02	0.289321	0.712926	3	176	76	0.026235	0.022874	−0.001687	−0.003729	0.002043	0.000616
98 − 02/03	−0.166092	−0.427295	4	104	148	0.012894	0.022808	0.000670	0.001181	−0.000511	−0.000300
99 − 03/04	−0.145091	−0.790353	4	159	97	0.010996	0.007673	−0.000563	0.001516	−0.002079	−0.000788
00 − 04/05	−0.100827	−0.729997	1	144	112	0.007020	0.008396	0.000840	0.000985	−0.000145	−0.000063
01 − 05/06	−0.011856	−0.073692	1	251	3	0.009756	0.006623	0.000713	0.004046	−0.003333	−0.000039
02 − 06/07	−0.141600	−1.150162	1	2	249	0.000646	0.009800	0.004023	0.000725	0.003298	0.003272
Panel B: Transaction costs 0.25%											
97 − 01/02	0.291417	0.841870	15	154	98	0.028948	0.018006	−0.001504	−0.003558	0.002054	0.000799
98 − 02/03	−0.024480	0.269521	1	140	112	0.013628	0.024699	0.001488	0.000323	0.001165	0.000518
99 − 03/04	−0.062704	−0.366026	9	196	60	0.010484	0.007787	0.000152	0.000460	−0.000308	−0.000072
00 − 04/05	−0.204957	−1.752057	0	0	256	0.000000	0.007637	0.000000	0.000903	−0.000903	−0.000903
01 − 05/06	−0.159469	−0.918730	1	4	250	0.011250	0.009722	0.000872	0.000750	0.000122	0.000120
02 − 06/07	−0.128586	−0.747523	1	45	206	0.009473	0.009851	0.000627	0.000778	−0.000151	−0.000124
Panel C: Transaction costs 0.5%											
97 − 01/02	0.546656	1.271963	1	99	153	0.013411	0.030541	−0.000539	−0.003444	0.002904	0.001763
98 − 02/03	−0.224627	−0.723953	3	87	165	0.012700	0.022042	0.000273	0.001338	−0.001065	−0.000697
99 − 03/04	−0.038918	−0.234982	1	198	58	0.010667	0.006726	0.000069	0.000753	−0.000684	−0.000155
00 − 04/05	−0.073023	−0.541169	1	188	68	0.007643	0.007671	0.000811	0.001160	−0.000349	−0.000093
01 − 05/06	−0.101090	−0.374219	1	54	200	0.006813	0.010379	0.001268	0.000613	0.000656	0.000516
02 − 06/07	−0.145598	−0.867518	1	51	200	0.009194	0.009924	0.000239	0.000882	−0.000642	−0.000512

Table 13.6a: 5-years training and 1-year out-of-sample DAX results. “Sample” denotes the training period used followed by the out-of-sample testing period (for example 97-99 implies that training data from 1997, 1998 and 1999 have been used to derive a trading rule which is then applied out-of-sample to data from 2000 and so on).“Excess” measures the fitness implied by a GP trading rule defined as excess return over a buy-and-hold strategy during the out-of-sample period, i.e. $(r_{gp} - r_{bh})$. ΔSOR indicates the excess Sortino ratio defined as $(SOR_{gp} - SOR_{bh})$ (both annualized) over the specified out-of-sample period. $\#T$ indicates the number of trades executed by a trading rule during out-of-sample testing with N_b and N_s denoting the number of buy-days (in-the-market) and sell-days (out-of-the-market), respectively. σ_b and σ_s indicate the standard deviation of returns during GP-in-market-days and GP-out-of-market-days, respectively. $\bar{r}_b$ and $\bar{r}_s$ denote the mean daily market return during GP-in-days and GP-out-days with $(\bar{r}_b - \bar{r}_s)$ as the difference between the two. $(\bar{r}_b - \bar{r}_m)$ measures the difference between mean daily returns during GP-in-days and buy-and-hold.

Sample	r_{gp}	r_{bh}	Δr	SOR_{gp}	SOR_{bh}	ΔSOR
Panel A: Transaction costs 0.1%						
97 − 01/02	−0.292976	−0.582296	0.289321	−0.757054	−1.469980	0.712926
98 − 02/03	0.076427	0.242520	−0.166092	0.368108	0.795403	−0.427295
99 − 03/04	−0.089651	0.055440	−0.145091	−0.464808	0.325546	−0.790353
00 − 04/05	0.128457	0.229284	−0.100827	1.050437	1.780435	−0.729997
01 − 05/06	0.177135	0.188991	−0.011856	1.023830	1.097522	−0.073692
02 − 06/07	0.044935	0.186534	−0.141600	0.000000	1.150162	−1.150162
Panel B: Transaction costs 0.25%						
97 − 01/02	−0.293879	−0.585296	0.291417	−0.635684	−1.477554	0.841870
98 − 02/03	0.215039	0.239520	−0.024480	1.054936	0.785415	0.269521
99 − 03/04	−0.010264	0.052440	−0.062704	−0.056820	0.309206	−0.366026
00 − 04/05	0.021327	0.226284	−0.204957	0.000000	1.752057	−1.752057
01 − 05/06	0.026522	0.185991	−0.159469	0.165954	1.084684	−0.918730
02 − 06/07	0.054948	0.183534	−0.128586	0.389227	1.136750	−0.747523
Panel C: Transaction costs 0.5%						
97 − 01/02	−0.043641	−0.590297	0.546656	−0.218213	−1.490176	1.271963
98 − 02/03	0.009892	0.234520	−0.224627	0.047880	0.771833	−0.723953
99 − 03/04	0.008522	0.047440	−0.038918	0.045796	0.280778	−0.234982
00 − 04/05	0.148261	0.221284	−0.073023	1.161483	1.702652	−0.541169
01 − 05/06	0.079901	0.180991	−0.101090	0.679623	1.053842	−0.374219
02 − 06/07	0.032936	0.178534	−0.145598	0.242606	1.110124	−0.867518

Table 13.6b: 5-years training and 1-year out-of-sample DAX results. "Sample" denotes the length of training and subsequent out-of-sample period. r_{gp} and r_{bh} denote the annualized out-of-sample returns for the GP trading rule and buy-and-hold, respectively. Δr is the difference between them. SOR_{gp} and SOR_{bh} indicate the repective annualized Sortino ratios for the GP trading rule and buy-and-hold, ΔSOR measures the difference between the two and is equal to ΔSOR in the preceding table.

Sample	Excess	ΔSOR	$\#T$	N_b	N_s	σ_b	σ_s	$\bar{r}_b$	$\bar{r}_s$	$(\bar{r}_b - \bar{r}_s)$	$(\bar{r}_b - \bar{r}_m)$
Panel A: Transaction costs 0.1%											
97 – 01/02 – 03	0.176556	0.243476	4	391	114	0.022935	0.022001	−0.000246	−0.001481	0.001236	0.000279
98 – 02/03 – 04	−0.164647	−0.215887	5	172	337	0.011035	0.017081	0.000748	0.000554	0.000194	0.000129
99 – 03/04 – 05	−0.165948	−0.564840	7	411	102	0.009112	0.007600	0.000328	0.001591	−0.001263	−0.000251
00 – 04/05 – 06	−0.270765	−0.759737	1	144	367	0.007020	0.009320	0.000840	0.000842	−0.000002	−0.000001
01 – 05/06 – 07	−0.011856	−0.037997	1	503	3	0.009756	0.006623	0.000756	0.004046	−0.003290	−0.000020
Panel B: Transaction costs 0.25%											
97 – 01/02 – 03	0.296283	0.411566	21	317	188	0.024523	0.019251	0.000346	−0.001992	0.002338	0.000870
98 – 02/03 – 04	−0.067249	−0.016616	1	148	361	0.013363	0.016027	0.001460	0.000275	0.001186	0.000841
99 – 03/04 – 05	−0.062704	−0.224032	9	453	60	0.008977	0.007787	0.000595	0.000460	0.000135	0.000016
00 – 04/05 – 06	−0.375042	−1.393211	0	0	511	0.000000	0.008727	0.000000	0.000841	−0.000841	−0.000841
01 – 05/06 – 07	−0.159469	−0.461590	1	256	250	0.009775	0.009722	0.000800	0.000750	0.000050	0.000025
Panel C: Transaction costs 0.5%											
97 – 01/02 – 03	0.361094	0.614920	2	142	363	0.012642	0.025601	0.000468	−0.000913	0.001381	0.000993
98 – 02/03 – 04	−0.267396	−0.520770	3	95	414	0.012336	0.015910	0.000332	0.000685	−0.000353	−0.000287
99 – 03/04 – 05	−0.250639	−0.837612	4	329	184	0.009818	0.006714	0.000186	0.001282	−0.001096	−0.000393
00 – 04/05 – 06	−0.242961	−0.680337	1	188	323	0.007643	0.009311	0.000811	0.000859	−0.000048	−0.000031
01 – 05/06 – 07	−0.101090	−0.245903	1	306	200	0.009313	0.010379	0.000881	0.000613	0.000269	0.000106

Table 13.7a: 5-years training and 2-years out-of-sample DAX results. "Sample" denotes the training period used followed by the out-of-sample testing period (for example 97-99 implies that training data from 1997, 1998 and 1999 have been used to derive a trading rule which is then applied out-of-sample to data from 2000 and so on)."Excess" measures the fitness implied by a GP trading rule defined as excess return over a buy-and-hold strategy during the out-of-sample period, i.e. $(r_{gp} - r_{bh})$. ΔSOR indicates the excess Sortino ratio defined as $(SOR_{gp} - SOR_{bh})$ (both annualized) over the specified out-of-sample period. $\#T$ indicates the number of trades executed by a trading rule during out-of-sample testing with N_b and N_s denoting the number of buy-days (in-the-market) and sell-days (out-of-the-market), respectively. σ_b and σ_s indicate the standard deviation of returns during GP-in-market-days and GP-out-of-market-days, respectively. $\bar{r}_b$ and $\bar{r}_s$ denote the mean daily market return during GP-in-days and GP-out-days with $(\bar{r}_b - \bar{r}_s)$ as the difference between the two. $(\bar{r}_b - \bar{r}_m)$ measures the difference between mean daily returns during GP-in-days and buy-and-hold.

Sample	r_{gp}	r_{bh}	Δr	SOR_{gp}	SOR_{bh}	ΔSOR
Panel A: Transaction costs 0.1%						
97 − 01/02 − 03	−0.045180	−0.133458	0.088278	−0.129894	−0.373370	0.243476
98 − 02/03 − 04	0.074337	0.156661	−0.082324	0.413899	0.629786	−0.215887
99 − 03/04 − 05	0.064536	0.147509	−0.082974	0.401612	0.966453	−0.564840
00 − 04/05 − 06	0.078597	0.213980	−0.135382	0.643977	1.403714	−0.759737
01 − 05/06 − 07	0.189177	0.195105	−0.005928	1.127281	1.165278	−0.037997
Panel B: Transaction costs 0.25%						
97 − 01/02 − 03	0.013183	−0.134958	0.148141	0.033999	−0.377567	0.411566
98 − 02/03 − 04	0.121537	0.155161	−0.033624	0.608340	0.624956	−0.016616
99 − 03/04 − 05	0.114657	0.146009	−0.031352	0.731808	0.955840	−0.224032
00 − 04/05 − 06	0.024959	0.212480	−0.187521	0.000000	1.393211	−1.393211
01 − 05/06 − 07	0.113870	0.193605	−0.079735	0.697409	1.158998	−0.461590
Panel C: Transaction costs 0.5%						
97 − 01/02 − 03	0.043089	−0.137458	0.180547	0.229628	−0.385292	0.614920
98 − 02/03 − 04	0.018963	0.152661	−0.133698	0.093868	0.614638	−0.520770
99 − 03/04 − 05	0.018190	0.143509	−0.125320	0.102417	0.940029	−0.837612
00 − 04/05 − 06	0.088499	0.209980	−0.121480	0.694666	1.375003	−0.680337
01 − 05/06 − 07	0.140560	0.191105	−0.050545	0.900511	1.146414	−0.245903

Table 13.7b: 5-years training and 2-years out-of-sample DAX results. "Sample" denotes the length of training and subsequent out-of-sample period. r_{gp} and r_{bh} denote the annualized out-of-sample returns for the GP trading rule and buy-and-hold, respectively. Δr is the difference between them. SOR_{gp} and SOR_{bh} indicate the repective annualized Sortino ratios for the GP trading rule and buy-and-hold, ΔSOR measures the difference between the two and is equal to ΔSOR in the preceding table.

Sample	Excess	ΔSOR	$\#T$	N_b	N_s	σ_b	σ_s	$\bar{r}_b$	$\bar{r}_s$	$(\bar{r}_b - \bar{r}_s)$	$(\bar{r}_b - \bar{r}_m)$
Panel A: Transaction costs 0.1%											
97 − 01/02 − 04	0.176556	0.187273	4	648	114	0.018870	0.022001	−0.000039	−0.001481	0.001442	0.000216
98 − 02/03 − 05	−0.295349	−0.292433	9	324	442	0.009368	0.015464	0.000731	0.000720	0.000011	0.000007
99 − 03/04 − 06	−0.303883	−0.654610	9	577	191	0.009362	0.008354	0.000327	0.001606	−0.001279	−0.000318
00 − 04/05 − 07	−0.432773	−0.804055	1	144	619	0.007020	0.009500	0.000840	0.000824	0.000016	0.000013
Panel B: Transaction costs 0.25%											
97 − 01/02 − 04	0.296283	0.316006	21	574	188	0.019381	0.019251	0.000314	−0.001992	0.002306	0.000569
98 − 02/03 − 05	−0.297818	−0.374835	5	219	547	0.011738	0.013784	0.001048	0.000595	0.000454	0.000324
99 − 03/04 − 06	−0.084620	−0.193984	11	706	62	0.009243	0.007840	0.000646	0.000641	0.000004	0.000000
00 − 04/05 − 07	−0.536703	−1.336389	0	0	763	0.000000	0.009080	0.000000	0.000827	−0.000827	−0.000827
Panel C: Transaction costs 0.5%											
97 − 01/02 − 04	0.244350	0.288827	2	287	475	0.011741	0.022773	0.000040	−0.000433	0.000472	0.000294
98 − 02/03 − 05	−0.485542	−0.727111	3	95	671	0.012336	0.013354	0.000332	0.000780	−0.000448	−0.000392
99 − 03/04 − 06	−0.327565	−0.714494	5	417	351	0.009582	0.008574	0.000414	0.000920	−0.000505	−0.000231
00 − 04/05 − 07	−0.404969	−0.759013	1	188	575	0.007643	0.009509	0.000811	0.000833	−0.000022	−0.000016

Table 13.8a: 5-years training and 3-years out-of-sample DAX results. “Sample” denotes the training period used followed by the out-of-sample testing period (for example 97-99 implies that training data from 1997, 1998 and 1999 have been used to derive a trading rule which is then applied out-of-sample to data from 2000 and so on).“Excess” measures the fitness implied by a GP trading rule defined as excess return over a buy-and-hold strategy during the out-of-sample period, i.e. $(r_{gp} - r_{bh})$. ΔSOR indicates the excess Sortino ratio defined as $(SOR_{gp} - SOR_{bh})$ (both annualized) over the specified out-of-sample period. $\#T$ indicates the number of trades executed by a trading rule during out-of-sample testing with N_b and N_s denoting the number of buy-days (in-the-market) and sell-days (out-of-the-market), respectively. σ_b and σ_s indicate the standard deviation of returns during GP-in-market-days and GP-out-of-market-days, respectively. $\bar{r}_b$ and $\bar{r}_s$ denote the mean daily market return during GP-in-days and GP-out-days with $(\bar{r}_b - \bar{r}_s)$ as the difference between the two. $(\bar{r}_b - \bar{r}_m)$ measures the difference between mean daily returns during GP-in-days and buy-and-hold.

Sample	r_{gp}	r_{bh}	Δr	SOR_{gp}	SOR_{bh}	ΔSOR
Panel A: Transaction costs 0.1%						
97 – 01/02 – 04	−0.006519	−0.065371	0.058852	−0.022019	−0.209292	0.187273
98 – 02/03 – 05	0.085851	0.184300	−0.098450	0.545398	0.837831	−0.292433
99 – 03/04 – 06	0.063270	0.164565	−0.101294	0.378159	1.032769	−0.654610
00 – 04/05 – 07	0.065468	0.209726	−0.144258	0.538867	1.342923	−0.804055
Panel B: Transaction costs 0.25%						
97 – 01/02 – 04	0.032390	−0.066371	0.098761	0.103234	−0.212772	0.316006
98 – 02/03 – 05	0.084028	0.183300	−0.099273	0.458129	0.832964	−0.374835
99 – 03/04 – 06	0.135358	0.163565	−0.028207	0.832319	1.026303	−0.193984
00 – 04/05 – 07	0.029825	0.208726	−0.178901	0.000000	1.336389	−1.336389
Panel C: Transaction costs 0.5%						
97 – 01/02 – 04	0.013412	−0.068038	0.081450	0.070718	−0.218109	0.288827
98 – 02/03 – 05	0.019786	0.181634	−0.161847	0.097623	0.824735	−0.727111
99 – 03/04 – 06	0.052710	0.161898	−0.109188	0.302407	1.016902	−0.714494
00 – 04/05 – 07	0.072070	0.207059	−0.134990	0.568299	1.327311	−0.759013

Table 13.8b: 5-years training and 3-years out-of-sample DAX results. "Sample" denotes the length of training and subsequent out-of-sample period. r_{gp} and r_{bh} denote the annualized out-of-sample returns for the GP trading rule and buy-and-hold, respectively. Δr is the difference between them. SOR_{gp} and SOR_{bh} indicate the repective annualized Sortino ratios for the GP trading rule and buy-and-hold, ΔSOR measures the difference between the two and is equal to ΔSOR in the preceding table.

	Trading Rule	*1-year out-of-sample*		*2-years out-of-sample*		*3-years out-of-sample*	
		Excess	ΔSOR	Excess	ΔSOR	Excess	ΔSOR
3-years in-sample	$97-99/...c05$	0.032598	0.135917	0.005863	0.011110		
	$99-01/...c05$	0.593067	1.496094	0.276050	0.386905	0.172612	0.184907
	$04-06/...c05$	0.016128	0.090852				
5-years in-sample	$97-01/...c025$	0.291417	0.841870	0.296283	0.411566	0.296238	0.316006
	$98-02/...c025$	−0.024480	0.269521				
	$97-01/...c05$	0.546656	1.271963	0.361094	0.614920	0.244350	0.288827

Table 13.9: Best Genetic Programming trading rules for the DAX.

13.3.2 Testing the Hang Seng

13.3.2.1 Test Results

Hang Seng results have been compiled in Tables 13.11a - 13.16a and adopt the already familiar table layout from the DAX scenarios. As already pointed out before, all scenarios under unrealistcally low transaction costs of $c = 0.1$ will not be addressed in-depth and only serve for comparative statics.

Two GP rules manage to outperform buy-and-hold under $c = 0.25$ in Table 13.11a. The first rule (97-99/00) is one of the rare instances where GP always stays out-of-the-market. This might reflect the aftermath of the Asian crisis starting in 1997 which coincides with the training period of the GP algorithm[98].

Rolling the time window forward by one year results once more in an outperformance compared to buy-and-hold, this time with a real timing strategy. Interestingly, the strategy starts with a prolonged out-of-the-market position and switches into the market shortly before Sept. 11th[99]. As this is the only trade the rule executes, it remains in-the-market even after Sept. 11th. Despite the unfortunate timing the rule yields just a small negative absolute return (-0.015305 p.a. see Table 13.11b) vs. a massive -0.270973 p.a. loss for buy-and-hold which at first sight seems puzzling. Some further analysis showed that the market was already down roughly 23% before Sept. 11th so this event did not add up much to the losses already incurred in the Hang Seng[100]. This explains the surprisingly good performance of the rule despite the, from an intuitive point of view, unfortunate timing[101]. Furthermore, volatility during buy-days is higher than during sell-days and timing abilities of the rule are insignificant. A replication of buy-and-hold is suggested for the 99-01/02 and 01-03/04 scenarios.

Focusing on the $c = 0.5$ panel, two rules emerge that beat buy-and-hold on

[98] The impact of the Asian crisis is easily spotted in Figure 13.2.

[99] As a general remark, GP investment positions can be easily superimposed on the respective time series using Matlab, however the author chose not to include these charts in order to save space.

[100] The heavy losses may be attributed to a massive outbreak of Severe Acute Respiratory Syndrome (SARS) also known as bird flu in 2001.

[101] As a sidenote, it is interesting that the DAX (and most likely all western stock indices) were far more affected by Sept. 11th than the Hang Seng.

a risk-adjusted basis. The 98-00/01c05 rule repeats the unfortunate timing of the rule discussed above (staying out and then switching in shortly before Sept. 11th) once more but switches out-of-the-market after 51 days rather than staying in until the end of the year. As most of the total loss in 2001 already occurred before Sept. 11th, the rule manages to cut losses considerably (-0.165865 p.a. vs. -0.275973 p.a., see Table 13.11b). Volatility during buy-days is once more higher than during sell days (0.025094 vs. 0.015039). The 99-01/02c05 rule yields positive results as well. It only takes a single in-the-market position in the mid of the trading year to cut losses considerably (-0.143000 vs. -0.206992 see Table 13.11b). Unfortunately, the difference $(\bar{r}_b - \bar{r}_s)$ and $(\bar{r}_b - \bar{r}_m)$ are in neither case statistically significant. Leaving aside the $c = 0.1$ rules, the remaining GP rules clearly underperform the benchmark. Similar to the results from the DAX, trading frequencies seems to be quite unaffected by transaction costs. The impact of transaction costs will become more visible in the later scenarios with a 5-year training horizon.

Stretching the horizon to two years out-of-sample yields similar results (Table 13.12a). The rules obtained during the 97-99 and 98-00 training sample outperform buy-and-hold in terms of excess return and excess Sortino ratio for both transaction costs scenarios ($c = 0.25$ and $c = 0.5$). Interestingly, the same rules execute at most a single trade during the first out-of-sample year (see Table 13.11a) whereas trading activity picks up during the second year. As already observed before, volatility during in-days is higher than during out-days.

Returning to Table 13.12a, some brief remarks on the $c = 0.1$ rules are in order. Despite the unrealistically low transaction cost, the algorithm did not find any successful trading rule in scenarios where the $c = 0.25$ and $c = 0.5$ rules failed as well to beat buy-and-hold. This might hint at market efficiency for these particular periods in the market since even with extremely low transaction costs (and thus the possibility of almost zero transaction costs), no technical trading rule could be found that beats the benchmark. This applies to both the 3:1 and the 3:2 scenarios illustrated in Tables 13.11a and 13.12a. The 03-05/06(-07) case is an exception that yields positive returns while the same scenario under higher transaction costs does not. However, as the rule is inexistent under realistic transaction costs, this does not contradict market efficiency. Speaking of the $c = 0.1$ rules in Table 13.12a, the 02-04/05-06 rule manages to forecast market returns at a statistically significant level ($\alpha = 0.05$) despite a marginally negative risk-adjusted performance.

As a final observation, it is worth mentioning that neither of the successful rules yields positive absolute returns p.a. (Table 13.12b). The rules rather seem to rely on their power to switch out-of-the-market at the "right time" to cut losses instead. However $(\bar{r}_b - \bar{r}_m)$ and $(\bar{r}_b - \bar{r}_s)$ are mostly insignificant in the cases discussed so far.

Stretching the out-of-sample horizon further, Table 13.13a is in line with the result discussed so far. Three rules are still successful for $c = 0.25$ and $c = 0.5$. Interestingly, σ_b is roughly equal to σ_s for these rules which has not been the case in shorter out-of-sample scenarios.

Extending the training periods to 5 years, things look slightly different (Table 13.14a). There are only two succesful rules in total, both of them in the $c = 0.25$ panel. The 97-01/02 rule yields a considerably better Sortino ratio as does the 00-04/05 rule. Both rules execute just a single trade and volatility is almost equal during buy- and sell-days. All other rules fail to beat a buy-and-hold strategy, even the $c = 0.1$ scenarios. As another observation, the negative correlation between transaction costs and trading frequency can be seen in Table 13.14a though the effect will become more visible later.

The results do not change much for a 2-year out-of-sample horizon (Table 13.15a). The only successful rules are the 99-03/04-05 (which was just buy-and-hold in Table 13.14a) and the 00-04/05-06 rule for $c = 0.25$ with the latter consisting of just a single trade. Volatility during buy-days is once more slightly higher than during sell-days. Another point is the now clearly negative relationship between transaction costs and trading frequencies.

Stretching the out-of-sample period to 3 years (Table 13.16a) yields two successful rules. The 97-01/02-04c025 rule comes back into positive territory (it yielded negative returns for 2-years out-of-sample) and the 00-04/05-07c025 rule is still profitable. However, a look at the Sortino ratio shows that outperformance is at most marginal for both rules. The other rule severly underperform the benchmark.

Summarizing the most important results for the Hang Seng, the following points can be made:

- GP-optimized rules largely fail at beating the buy-and-hold benchmark on a risk-adjusted basis...
- but several rules outperform the benchmark during the years 2000-2002 (technology bubble burst, bird flu, Sept. 11th)
- one rule outperforms the benchmark in later years despite a sustained rise in the index (which favours buy-and-hold)
- the rules have no statistically significant forecasting power.

A list of successful trading rules for the Hang Seng has been compiled in Table 13.17. As usual, the $c = 0.1$ rules will not be addressed further. In general, the power of the GP rules seems to decline over time both in terms of excess returns and excess Sortino ratios. Therefore, as pointed out before in the case of the DAX, a 1-year out-of-sample period seems to work best when using GP. Concerning the 00-04/...c025 rule, it might seem puzzling as to why the excess return remains the same over all out-of-sample periods. Upon closer inspection of the rule (see Tables 13.14a-13.16a) it turns out that it executes a single deal during the first out-of-sample year and simply stays in-the-market afterwards when the out-of-sample period is extended to up to 3 years. The singular trade in the first year outperforms the benchmark earning 0.053836 and this return is carried throughout the subsequent periods during which the rule simply takes a sustained buy-and-hold position which does not add or subtract anything from the first year returns. Therefore, performance in excess of buy-and-hold remains the same. As already pointed out before, most of the rules (6 out of 8) beat the market during the years 2000-2002 where the market retreated due to the burst of the dot-com bubble, bird flu in Hong Kong (which probably had the hardest impact on the Hang Seng) and the events of Sept. 11th.

13.3.2.2 Structure of Trading Rules

A set of successful rules for the Hang Seng is illustrated in Figure 13.6. The set is not exhaustive (see Table 13.17) since some rules are quite complex and do not have an easy-to-grasp economic interpretation. First of all, all rules shown have a surprisingly easy structure. A particularly easy rule was obtained during the 97-99c025 in-sample period (Figure 13.6a). The rule simply checks whether the closing price lagged by 200 days is less than the closing price lagged by 250 days[102]. If this is true, an in-position is taken, else the rules stays out-of-the-market.

The second tree (Figure 13.6b) depicts the rule obtained from the 98-00c025 sample and first checks whether the minimum over the last 150 trading days is less than the closing price 150 days ago and then checks whether the minimum over the last 200 trading days is less than the result from the aforementioned subtree (either *0=false* or *1=true*). If the rule evaluates as true, an in-the-market position is set up, else the rule stays out-of-the market. A mirrored version of this rule is shown in Figure 13.6c depicting the rule obtained for the 99-01c05 sample. It first checks whether the maximum over the last 50 trading days is greater than 1.02 and then checks whether the result from the subtree (either 0 or 1) is greater than the 150-day moving average.

Figure 13.6d features the boolean operator “and”. Basically, the “and” operator evaluates as *1=true* as long as both arguments related to it are both true. Therefore, the rule first checks whether the closing price lagged by 200 days is less than the closing price lagged by 100 days. Once more the result from this subtree is either *0=false* or *1=true*. The rule is in the market only if the sub-

[102] As a reminder, all price data used in this study have been normalized by dividing the closing price by its respective 250-day moving average. All indicators have been derived from normalized prices. Therefore, when speaking of prices, moving averages etc. the respective indicators based on *normalized* prices rather than the original data is meant.

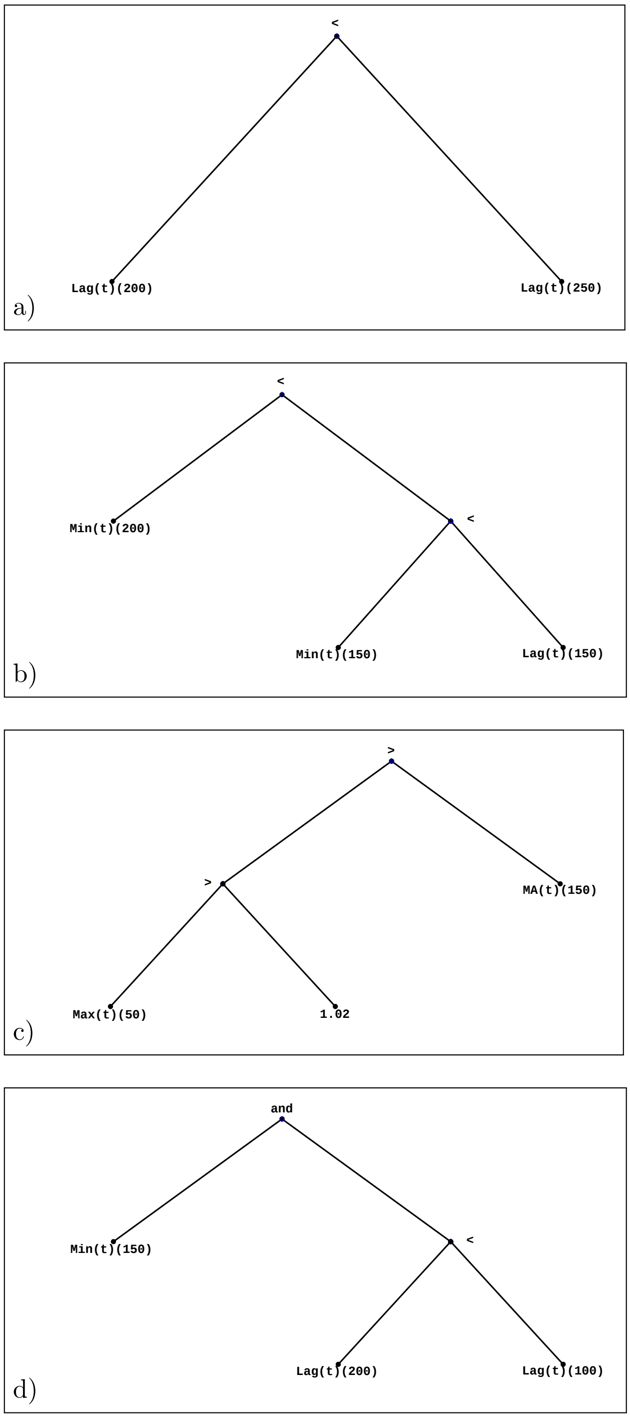


Figure 13.6: Tree structure of successful Hang Seng trading rules.

tree yields 1 and the left-hand side `min150` is $\neq 0$, else the rule stays out-of-the market[103].

As a final observation, it is noteworthy that the rules illustrated in Figure 13.6 have a tendency to pick up long term indicators (100, 150, 200 and 250 trading days time span) as was the case for the DAX trading rules[104]. These building blocks might imply that the GP algorithm relies on long-term trends in the market rather than reacting to short-lived (white) noise. The noticeable presence of long-term indicators in successful trading rules might also imply that technical trading rules should be generally based on long- rather than short-term variables.

13.3.2.3 Long Term Genetic Programming Performance

For the the sake of completeness, equity curves for 3:1 and 5:1 revolving GP strategies in the Hang Seng are illustrated in Figure 13.7. It is easily seen that GP fails to consistently beat the benchmark in all cases except the 3:1c025 scenario. However, one has to bear in mind that the Hang Seng showed a strong and sustained upward trend throughout the last couple of years as seen in Figure 13.2 making it very hard for GP to beat buy-and-hold[105]. Returning to the 3:1c025 scenario in Figure 13.7, it is remarkable how well GP stays above the benchmark. It partly avoids severe losses during the years 2000-2002 and still manages to keep its head above water in the following years. The tide finally turns against GP in 2007 when the benchmark continues to rise in a sustained upward trend with the benchmark overtaking GP. The most important question arising from this picture is of course whether the EMH still holds. To check this, summary statistics for the scenario are provided in Table 13.10.

It is noteworthy that GP (possibly due to some prolonged out-of-the-market periods) results in lower volatility (0.0078 vs. 0.0136) but higher skewness in absolute terms and a way higher excess kurtosis compared to the benchmark.

[103] The case `min150=0` occurs during the first 149 trading days as `min150` has not been initialized yet.

[104] The same applies to the more complex rules that have not been illustrated in Figure 13.6.

[105] This is of course a direct consequence of the choice of buy-and-hold as benchmark. A different benchmark might have resulted in a more favorable outcome for GP.

In terms of total return, GP lacks behind buy-and-hold (0.2846 vs. 0.4498) but most importantly, it is on par with the benchmark in terms of Sortino ratio. Therefore, it may be concluded that the Hang Seng was overall efficient during the years 2000-2007.

As a last exercise, the related kernel density estimates for the equity curves are shown in Figure 13.8. Three out of four scenarios feature a high spike around zero mirroring prolonged out-of-the-market positions (which tend to gather many tiny positive returns from the money market) and very small GP in-market returns. Only the 3:1c05 scenario spreads out a little more but still finishes well below the benchmark in terms of total return and Sortino Ratio (statistics not shown).

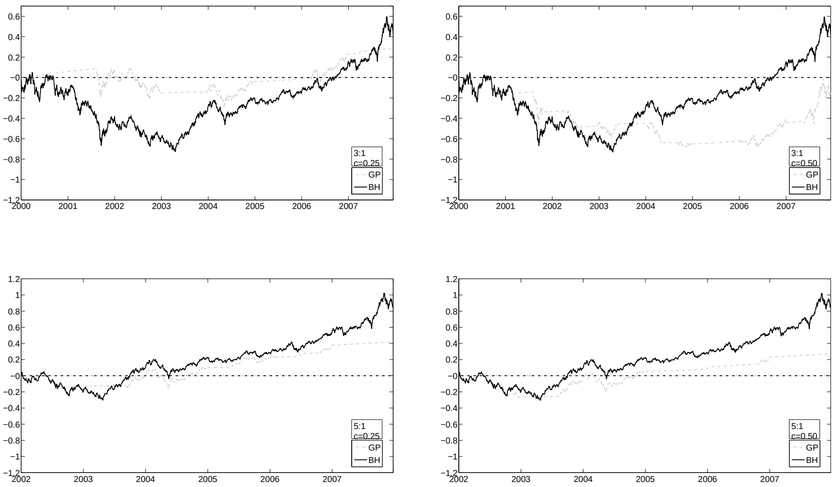


Figure 13.7: *Equity curves for 3:1 and 5:1 revolving Genetic Programming strategies for the Hang Seng for c=0.25 and c=0.5.*

	GP	*BH*
Sample	2000-2007	2000-2007
Mean	0.00014483	0.00022809
Median	0.00015840	0.00046297
Minimum	-0.0929	-0.0929
Maximum	0.0422	0.0576
Std.Dev.	0.0078	0.0136
Skewness	-0.7971	-0.35792
Ex.Kurtosis	15.825	3.503
Total Return	0.2846	0.4498
Sortino Ratio	0.0256	0.0234

Table 13.10: Hang Seng 3:1 $c = 0.25$ revolving strategy results.

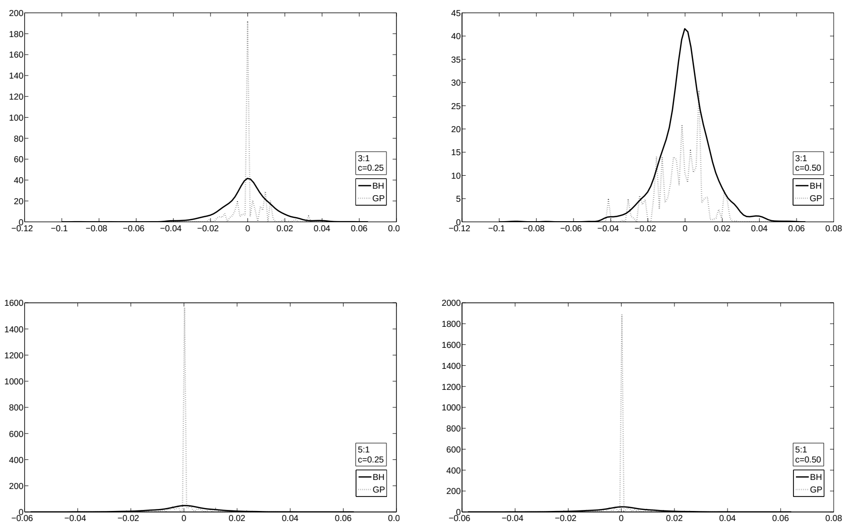


Figure 13.8: *Kernel smoothing density estimates for 3:1 and 5:1 Hang Seng scenarios for c=0.25 and c=0.5.*

Sample	Excess	ΔSOR	$\#T$	N_b	N_s	σ_b	σ_s	$\bar{r}_b$	$\bar{r}_s$	$(\bar{r}_b - \bar{r}_s)$	$(\bar{r}_b - \bar{r}_m)$
Panel A: Transaction costs 0.1%											
97 – 99/00	0.019813	0.079968	1	179	67	0.020146	0.018723	−0.000758	−0.000068	−0.000691	−0.000188
98 – 00/01	0.051160	0.286270	2	151	91	0.019557	0.013755	−0.001507	−0.000421	−0.001086	−0.000408
99 – 01/02	0.000000	0.000000	1	246	0	0.012209	0.000000	−0.000809	0.000000	−0.000809	0.000000
00 – 02/03	−0.263177	−1.723528	3	71	176	0.011694	0.010264	0.000417	0.001507	−0.001090	−0.000777
01 – 03/04	−0.101171	−0.660709	0	0	248	0.000000	0.010277	0.000000	−0.000577	0.000577	−0.000419
02 – 04/05	−0.015833	−0.331433	0	0	246	0.000000	0.007258	0.000000	0.000178	−0.000178	−0.000178
03 – 05/06	−0.077375	−0.558935	1	211	35	0.009196	0.008651	0.000980	0.002369	−0.001389	−0.000198
04 – 06/07	0.017007	−0.102322	1	145	99	0.019292	0.011498	0.002063	−0.000008	0.002071	0.000840
Panel B: Transaction costs 0.25%											
97 – 99/00	0.202731	0.446743	0	0	246	0.000000	0.019733	0.000000	−0.000570	0.000570	0.000570
98 – 00/01	0.255669	0.935284	1	92	150	0.020765	0.015361	−0.000407	−0.001523	0.001116	0.000692
99 – 01/02	0.000000	0.000000	1	246	0	0.012213	0.000000	−0.000801	0.000000	−0.000801	0.000000
00 – 02/03	−0.279931	−1.896715	0	0	247	0.000000	0.010681	0.000000	0.001193	−0.001193	−0.001193
01 – 03/04	0.000000	0.000000	1	248	0	0.010278	0.000000	0.000427	0.000000	0.000427	0.000000
02 – 04/05	−0.012833	−0.305865	0	0	246	0.000000	0.007258	0.000000	0.000178	−0.000178	−0.000178
03 – 05/06	−0.043383	−0.359742	1	196	50	0.009542	0.007293	0.001217	0.001023	0.000194	0.000039
04 – 06/07	−0.236797	−1.106381	1	1	243	0.000000	0.016569	0.020272	0.001144	0.019128	0.019049
Panel C: Transaction costs 0.5%											
97 – 99/00	−0.017500	−0.020753	1	180	66	0.020271	0.018292	−0.000960	0.000492	−0.001452	−0.000390
98 – 00/01	0.110108	0.590780	1	51	191	0.024961	0.015025	−0.003656	−0.000416	−0.003239	−0.002557
99 – 01/02	0.063992	0.378851	1	63	183	0.011297	0.012499	−0.002317	−0.000279	−0.002039	−0.001516
00 – 02/03	−0.259144	−1.696425	1	102	145	0.010989	0.010452	0.000306	0.001817	−0.001511	−0.000887
01 – 03/04	−0.299134	−1.822383	3	130	118	0.009743	0.010539	−0.001340	0.002373	−0.003713	−0.001766
02 – 04/05	−0.007833	−0.263583	0	0	246	0.000000	0.007258	0.000000	0.000178	−0.000178	−0.000178
03 – 05/06	−0.079027	−0.530141	3	225	21	0.009324	0.006385	0.001010	0.002965	−0.001955	−0.000167
04 – 06/07	−0.040182	−0.280656	2	146	98	0.019304	0.011432	0.001726	0.000473	0.001253	0.000503

Table 13.11a: 3-years training and 1-year out-of-sample Hang Seng results. "Sample" denotes the training period used followed by the out-of-sample testing period (for example 97-99 implies that training data from 1997, 1998 and 1999 have been used to derive a trading rule which is then applied out-of-sample to data from 2000 and so on). "Excess" measures the fitness implied by a GP trading rule defined as excess return over a buy-and-hold strategy during the out-of-sample period, i.e. $(r_{gp} - r_{bh})$. ΔSOR indicates the excess Sortino ratio defined as $(SOR_{gp} - SOR_{bh})$ (both annualized) over the specified out-of-sample period. $\#T$ indicates the number of trades executed by a trading rule during out-of-sample testing with N_b and N_s denoting the number of buy-days (in-the-market) and sell-days (out-of-the-market), respectively. σ_b and σ_s indicate the standard deviation of returns during GP-in-market-days and GP-out-of-market-days, respectively. $\bar{r}_b$ and $\bar{r}_s$ denote the mean daily market return during GP-in-days and GP-out-days with $(\bar{r}_b - \bar{r}_s)$ as the difference between the two. $(\bar{r}_b - \bar{r}_m)$ measures the difference between mean daily returns during GP-in-days and buy-and-hold.

Sample	r_{gp}	r_{bh}	Δr	SOR_{gp}	SOR_{bh}	ΔSOR
Panel A: Transaction costs 0.1%						
97 − 99/00	−0.122512	−0.142325	0.019813	−0.357786	−0.437753	0.079968
98 − 00/01	−0.216813	−0.267973	0.051160	−0.682543	−0.968813	0.286270
99 − 01/02	−0.198992	−0.198992	0.000000	−1.085768	−1.085768	0.000000
00 − 02/03	0.029574	0.292751	−0.263177	0.192825	1.916353	−1.723528
01 − 03/04	0.002630	0.103801	−0.101171	0.000000	0.660709	−0.660709
02 − 04/05	0.026071	0.041904	−0.015833	0.000000	0.331433	−0.331433
03 − 05/06	0.210231	0.287605	−0.077375	1.328138	1.887073	−0.558935
04 − 06/07	0.313355	0.296347	0.017007	1.016372	1.118694	−0.102322
Panel B: Transaction costs 0.25%						
97 − 99/00	0.057406	−0.145325	0.202731	0.000000	−0.446743	0.446743
98 − 00/01	−0.015305	−0.270973	0.255669	−0.043496	−0.978780	0.935284
99 − 01/02	−0.201992	−0.201992	0.000000	−1.102137	−1.102137	0.000000
00 − 02/03	0.009820	0.289751	−0.279931	0.000000	1.896715	−1.896715
01 − 03/04	0.100801	0.100801	0.000000	0.641614	0.641614	0.000000
02 − 04/05	0.026071	0.038904	−0.012833	0.000000	0.305865	−0.305865
03 − 05/06	0.241223	0.284605	−0.043383	1.506578	1.866321	−0.359742
04 − 06/07	0.056550	0.293347	−0.236797	0.000000	1.106381	−1.106381
Panel C: Transaction costs 0.5%						
97 − 99/00	−0.167825	−0.150325	−0.017500	−0.482417	−0.461664	−0.020753
98 − 00/01	−0.165865	−0.275973	0.110108	−0.404343	−0.995123	0.590780
99 − 01/02	−0.143000	−0.206992	0.063992	−0.750568	−1.129419	0.378851
00 − 02/03	0.025606	0.284751	−0.259144	0.175211	1.871636	−1.696425
01 − 03/04	−0.203333	0.095801	−0.299134	−1.210163	0.612220	−1.822383
02 − 04/05	0.026071	0.033904	−0.007833	0.000000	0.263583	−0.263583
03 − 05/06	0.200578	0.279605	−0.079027	1.300674	1.830815	−0.530141
04 − 06/07	0.248166	0.288347	−0.040182	0.805080	1.085736	−0.280656

Table 13.11b: 3-years training and 1-year out-of-sample Hang Seng results. “Sample” denotes the length of training and subsequent out-of-sample period. r_{gp} and r_{bh} denote the annualized out-of-sample returns for the GP trading rule and buy-and-hold, respectively. Δr is the difference between them. SOR_{gp} and SOR_{bh} indicate the repective annualized Sortino ratios for the GP trading rule and buy-and-hold, ΔSOR measures the difference between the two and is equal to ΔSOR in the preceding table.

Sample	Excess	ΔSOR	$\#T$	N_b	N_s	σ_b	σ_s	$\bar{r}_b$	$\bar{r}_s$	$(\bar{r}_b - \bar{r}_s)$	$(\bar{r}_b - \bar{r}_m)$
Panel A: Transaction costs 0.1%											
97 − 99/00 − 01	0.338323	0.578051	1	179	310	0.020146	0.017796	−0.000758	−0.000921	0.000163	0.000103
98 − 00/01 − 02	0.051160	0.138828	2	398	91	0.015385	0.013755	−0.001077	−0.000421	−0.000656	−0.000122
99 − 01/02 − 03	−0.246153	−0.698253	3	260	234	0.012054	0.010795	−0.000572	0.001073	−0.001645	−0.000779
00 − 02/03 − 04	−0.384107	−1.235833	3	71	425	0.011694	0.010294	0.000417	0.000915	−0.000498	−0.000427
01 − 03/04 − 05	−0.155408	−0.542573	4	21	474	0.006567	0.008974	−0.001264	0.000373	−0.001637	−0.001568
02 − 04/05 − 06	−0.024109	−0.004012	1	139	354	0.008538	0.008079	0.001945	0.000191	0.001754*	0.001260
03 − 05/06 − 07	0.027383	0.240005	3	399	92	0.012280	0.017278	0.001560	−0.000189	0.001749	0.000328
Panel B: Transaction costs 0.25%											
97 − 99/00 − 01	0.067099	0.146903	8	142	347	0.019707	0.018202	−0.002749	−0.000089	−0.002660	−0.001888
98 − 00/01 − 02	0.248648	0.542211	9	322	167	0.015161	0.014961	−0.000642	−0.001559	0.000917	0.000313
99 − 01/02 − 03	−0.253740	−0.711241	1	249	245	0.012251	0.010622	−0.000645	0.001074	−0.001719	−0.000853
00 − 02/03 − 04	−0.400861	−1.330793	0	0	496	0.000000	0.010495	0.000000	0.000843	−0.000843	−0.000843
01 − 03/04 − 05	0.000000	0.000000	1	495	0	0.008886	0.000000	0.000303	0.000000	0.000303	0.000000
02 − 04/05 − 06	−0.048164	−0.257719	1	243	250	0.009110	0.007294	0.001083	0.000300	0.000784	0.000397
03 − 05/06 − 07	−0.043383	−0.144309	1	441	50	0.013883	0.007293	0.001256	0.001023	0.000233	0.000024
Panel C: Transaction costs 0.5%											
97 − 99/00 − 01	0.301239	0.523493	2	205	284	0.019616	0.017992	−0.000781	−0.000920	0.000139	0.000081
98 − 00/01 − 02	0.058290	0.194659	5	221	268	0.015867	0.014396	−0.001833	−0.000231	−0.001601	−0.000878
99 − 01/02 − 03	−0.080744	−0.237490	2	139	355	0.010961	0.011709	0.000089	0.000254	−0.000165	−0.000119
00 − 02/03 − 04	−0.379975	−1.220762	1	102	394	0.010989	0.010373	0.000306	0.000982	−0.000676	−0.000537
01 − 03/04 − 05	−0.349152	−1.188477	4	348	147	0.008303	0.009906	−0.000501	0.002208	−0.002709	−0.000805
02 − 04/05 − 06	−0.048164	−0.253549	1	243	250	0.009110	0.007294	0.001083	0.000300	0.000784	0.000397
03 − 05/06 − 07	−0.079027	−0.185315	3	470	21	0.013583	0.006385	0.001155	0.002965	−0.001810	−0.000077

Table 13.12a: 3-years training and 2-years out-of-sample Hang Seng results. "Sample" denotes the training period used followed by the out-of-sample testing period (for example 97-99 implies that training data from 1997, 1998 and 1999 have been used to derive a trading rule which is then applied out-of-sample to data from 2000 and so on). "Excess" measures the fitness implied by a GP trading rule defined as excess return over a buy-and-hold strategy during the out-of-sample period, i.e. $(r_{gp} - r_{bh})$. ΔSOR indicates the excess Sortino ratio defined as $(SOR_{gp} - SOR_{bh})$ (both annualized) over the specified out-of-sample period. $\#T$ indicates the number of trades executed by a trading rule during out-of-sample testing with N_b and N_s denoting the number of buy-days (in-the-market) and sell-days (out-of-the-market), respectively. σ_b and σ_s indicate the standard deviation of returns during GP-in-market-days and GP-out-of-market-days, respectively. $\bar{r}_b$ and $\bar{r}_s$ denote the mean daily market return during GP-in-days and GP-out-days with $(\bar{r}_b - \bar{r}_s)$ as the difference between the two. $(\bar{r}_b - \bar{r}_m)$ measures the difference between mean daily returns during GP-in-days and buy-and-hold. (*) indicates significance for $\alpha = 0.05$.

Sample	r_{gp}	r_{bh}	Δr	SOR_{gp}	SOR_{bh}	ΔSOR
Panel A: Transaction costs 0.1%						
97 − 99/00 − 01	−0.042516	−0.211677	0.169161	−0.124926	−0.702977	0.578051
98 − 00/01 − 02	−0.208940	−0.234520	0.025580	−0.859465	−0.998293	0.138828
99 − 01/02 − 03	−0.072830	0.050246	−0.123077	−0.402734	0.295519	−0.698253
00 − 02/03 − 04	0.016110	0.208164	−0.192053	0.104619	1.340452	−1.235833
01 − 03/04 − 05	−0.003595	0.074109	−0.077704	−0.026967	0.515606	−0.542573
02 − 04/05 − 06	0.155992	0.168047	−0.012054	1.198812	1.202824	−0.004012
03 − 05/06 − 07	0.315245	0.301554	0.013691	1.614789	1.374784	0.240005
Panel B: Transaction costs 0.25%						
97 − 99/00 − 01	−0.179628	−0.213177	0.033550	−0.560794	−0.707697	0.146903
98 − 00/01 − 02	−0.111697	−0.236020	0.124324	−0.461956	−1.004167	0.542211
99 − 01/02 − 03	−0.078123	0.048746	−0.126870	−0.424543	0.286697	−0.711241
00 − 02/03 − 04	0.006233	0.206664	−0.200430	0.000000	1.330793	−1.330793
01 − 03/04 − 05	0.072609	0.072609	0.000000	0.504660	0.504660	0.000000
02 − 04/05 − 06	0.142465	0.166546	−0.024082	0.932328	1.190046	−0.257719
03 − 05/06 − 07	0.278363	0.300054	−0.021691	1.222697	1.367006	−0.144309
Panel C: Transaction costs 0.5%						
97 − 99/00 − 01	−0.065058	−0.215677	0.150620	−0.191990	−0.715483	0.523493
98 − 00/01 − 02	−0.209375	−0.238520	0.029145	−0.819205	−1.013864	0.194659
99 − 01/02 − 03	0.005875	0.046246	−0.040372	0.035031	0.272521	−0.237490
00 − 02/03 − 04	0.014176	0.204164	−0.189988	0.096609	1.317372	−1.220762
01 − 03/04 − 05	−0.104467	0.070109	−0.174576	−0.702138	0.486340	−1.188477
02 − 04/05 − 06	0.139965	0.164046	−0.024082	0.914609	1.168158	−0.253549
03 − 05/06 − 07	0.258040	0.297554	−0.039514	1.168584	1.353899	−0.185315

Table 13.12b: 3-years training and 2-years out-of-sample Hang Seng results. "Sample" denotes the length of training and subsequent out-of-sample period. r_{gp} and r_{bh} denote the annualized out-of-sample returns for the GP trading rule and buy-and-hold, respectively. Δr is the difference between them. SOR_{gp} and SOR_{bh} indicate the repective annualized Sortino ratios for the GP trading rule and buy-and-hold, ΔSOR measures the difference between the two and is equal to ΔSOR in the preceding table.

Sample	Excess	ΔSOR	$\#T$	N_b	N_s	σ_b	σ_s	$\bar{r}_b$	$\bar{r}_s$	$(\bar{r}_b - \bar{r}_s)$	$(\bar{r}_b - \bar{r}_m)$
Panel A: Transaction costs 0.1%											
97 − 99/00 − 02	0.537661	0.694275	2	183	553	0.019948	0.015594	−0.000834	−0.000850	0.000016	0.000012
98 − 00/01 − 03	0.051160	0.081862	2	646	91	0.013801	0.013755	−0.000200	−0.000421	0.000221	0.000027
99 − 01/02 − 04	−0.412605	−0.800586	4	316	427	0.011774	0.010549	−0.000608	0.000979	−0.001587	−0.000912
00 − 02/03 − 05	−0.401262	−0.915326	3	71	672	0.011694	0.009291	0.000417	0.000645	−0.000228	−0.000206
01 − 03/04 − 06	−0.410190	−0.925857	4	21	721	0.006567	0.009020	−0.001264	0.000653	−0.001918	−0.001863
02 − 04/05 − 07	−0.295541	−0.257449	1	151	587	0.008951	0.012278	0.001824	0.000644	0.001180	0.000938
Panel B: Transaction costs 0.25%											
97 − 99/00 − 02	0.088498	0.097996	12	284	452	0.016380	0.016988	−0.001964	−0.000143	−0.001821	−0.001118
98 − 00/01 − 03	0.187040	0.291989	14	557	180	0.013492	0.014652	0.000101	−0.001242	0.001342	0.000328
99 − 01/02 − 04	−0.374683	−0.721055	1	249	494	0.012251	0.010463	−0.000645	0.000783	−0.001428	−0.000950
00 − 02/03 − 05	−0.418016	−1.036792	0	0	743	0.000000	0.009538	0.000000	0.000623	−0.000623	−0.000623
01 − 03/04 − 06	0.000000	0.000000	1	742	0	0.008962	0.000000	0.000599	0.000000	0.000599	0.000000
02 − 04/05 − 07	−0.048164	−0.208595	1	488	250	0.013377	0.007294	0.001186	0.000300	0.000886	0.000300
Panel C: Transaction costs 0.5%											
97 − 99/00 − 02	0.277910	0.360544	4	361	375	0.016645	0.016906	−0.001027	−0.000671	−0.000356	−0.000181
98 − 00/01 − 03	−0.300708	−0.400234	21	325	412	0.014379	0.013289	−0.000959	0.000349	−0.001308	−0.000731
99 − 01/02 − 04	−0.201687	−0.405572	2	139	604	0.010961	0.011148	0.000089	0.000354	−0.000265	−0.000216
00 − 02/03 − 05	−0.397131	−0.897709	1	102	641	0.010989	0.009295	0.000306	0.000673	−0.000367	−0.000316
01 − 03/04 − 06	−0.420245	−0.956280	5	572	170	0.008566	0.010024	0.000096	0.002293	−0.002197	−0.000503
02 − 04/05 − 07	−0.048164	−0.206508	1	488	250	0.013377	0.007294	0.001186	0.000300	0.000886	0.000300

Table 13.13a: 3-years training and 3-years out-of-sample Hang Seng results. "Sample" denotes the training period used followed by the out-of-sample testing period (for example 97-99 implies that training data from 1997, 1998 and 1999 have been used to derive a trading rule which is then applied out-of-sample to data from 2000 and so on). "Excess" measures the fitness implied by a GP trading rule defined as excess return over a buy-and-hold strategy during the out-of-sample period, i.e. $(r_{gp} - r_{bh})$. ΔSOR indicates the excess Sortino ratio defined as $(SOR_{gp} - SOR_{bh})$ (both annualized) over the specified out-of-sample period. $\#T$ indicates the number of trades executed by a trading rule during out-of-sample testing with N_b and N_s denoting the number of buy-days (in-the-market) and sell-days (out-of-the-market), respectively. σ_b and σ_s indicate the standard deviation of returns during GP-in-market-days and GP-out-of-market-days, respectively. $\bar{r}_b$ and $\bar{r}_s$ denote the mean daily market return during GP-in-days and GP-out-days with $(\bar{r}_b - \bar{r}_s)$ as the difference between the two. $(\bar{r}_b - \bar{r}_m)$ measures the difference between mean daily returns during GP-in-days and buy-and-hold.

Sample	r_{gp}	r_{bh}	Δr	SOR_{gp}	SOR_{bh}	ΔSOR
Panel A: Transaction costs 0.1%						
97 − 99/00 − 02	−0.028920	−0.208141	0.179220	−0.085875	−0.780150	0.694275
98 − 00/01 − 03	−0.039465	−0.056519	0.017053	−0.182741	−0.264603	0.081862
99 − 01/02 − 04	−0.062845	0.074690	−0.137535	−0.351021	0.449565	−0.800586
00 − 02/03 − 05	0.019827	0.153581	−0.133754	0.128927	1.044253	−0.915326
01 − 03/04 − 06	0.010739	0.147469	−0.136730	0.080596	1.006452	−0.925857
02 − 04/05 − 07	0.118685	0.217199	−0.098514	0.867489	1.124938	−0.257449
Panel B: Transaction costs 0.25%						
97 − 99/00 − 02	−0.179642	−0.209141	0.029499	−0.685708	−0.783704	0.097996
98 − 00/01 − 03	0.004828	−0.057519	0.062347	0.022819	−0.269169	0.291989
99 − 01/02 − 04	−0.051205	0.073690	−0.124894	−0.277510	0.443545	−0.721055
00 − 02/03 − 05	0.013242	0.152581	−0.139339	0.000000	1.036792	−1.036792
01 − 03/04 − 06	0.146469	0.146469	0.000000	0.999439	0.999439	0.000000
02 − 04/05 − 07	0.200144	0.216199	−0.016055	0.909976	1.118571	−0.208595
Panel C: Transaction costs 0.5%						
97 − 99/00 − 02	−0.118171	−0.210807	0.092637	−0.429047	−0.789591	0.360544
98 − 00/01 − 03	−0.159422	−0.059185	−0.100236	−0.677341	−0.277107	−0.400234
99 − 01/02 − 04	0.004794	0.072023	−0.067229	0.028511	0.434083	−0.405572
00 − 02/03 − 05	0.018538	0.150914	−0.132377	0.126502	1.024210	−0.897709
01 − 03/04 − 06	0.004720	0.144802	−0.140082	0.031304	0.987584	−0.956280
02 − 04/05 − 07	0.198477	0.214532	−0.016055	0.901255	1.107763	−0.206508

Table 13.13b: 3-years training and 3-years out-of-sample Hang Seng results. "Sample" denotes the length of training and subsequent out-of-sample period. r_{gp} and r_{bh} denote the annualized out-of-sample returns for the GP trading rule and buy-and-hold, respectively. Δr is the difference between them. SOR_{gp} and SOR_{bh} indicate the repective annualized Sortino ratios for the GP trading rule and buy-and-hold, ΔSOR measures the difference between the two and is equal to ΔSOR in the preceding table.

Sample	Excess	ΔSOR	$\#T$	N_b	N_s	σ_b	σ_s	$\bar{r}_b$	$\bar{r}_s$	$(\bar{r}_b - \bar{r}_s)$	$(\bar{r}_b - \bar{r}_m)$
Panel A: Transaction costs 0.1%											
97 − 01/02	−0.031945	−0.155294	3	132	114	0.011654	0.012790	−0.001764	0.000314	−0.002078	−0.000963
98 − 02/03	−0.172503	−1.266714	1	63	184	0.011471	0.010422	0.001797	0.000987	0.000810	0.000604
99 − 03/04	−0.203644	−1.196786	3	110	138	0.011101	0.009488	−0.000864	0.001455	−0.002319	−0.001291
00 − 04/05	−0.071645	−0.565159	10	139	107	0.007427	0.007044	−0.000159	0.000616	−0.000775	−0.000337
01 − 05/06	−0.247980	−1.887073	0	0	246	0.000000	0.009116	0.000000	0.001177	−0.001177	−0.001177
02 − 06/07	−0.255342	−1.118694	0	0	244	0.000000	0.016580	0.000000	0.001223	−0.001223	−0.001223
Panel B: Transaction costs 0.25%											
97 − 01/02	0.072829	0.402861	1	149	97	0.012141	0.012385	−0.000877	−0.000683	−0.000194	−0.000077
98 − 02/03	−0.167336	−1.238700	1	78	169	0.011665	0.010228	0.001525	0.001040	0.000485	0.000332
99 − 03/04	0.000000	0.000000	1	248	0	0.010278	0.000000	0.000427	0.000000	0.000427	0.000000
00 − 04/05	0.053836	0.444115	1	152	94	0.007317	0.007146	0.000609	−0.000517	0.001126	0.000430
01 − 05/06	−0.105882	−0.614069	3	77	169	0.009081	0.009124	0.002155	0.000732	0.001424	0.000978
02 − 06/07	−0.252342	−1.106381	0	0	244	0.000000	0.016580	0.000000	0.001223	−0.001223	−0.001223
Panel C: Transaction costs 0.5%											
97 − 01/02	−0.058052	−0.298035	2	202	44	0.011968	0.013248	−0.001228	0.001162	−0.002390	−0.000427
98 − 02/03	−0.076184	−0.641202	1	97	150	0.010911	0.010518	0.002175	0.000559	0.001616	0.000981
99 − 03/04	0.000000	0.000000	1	248	0	0.010278	0.000000	0.000427	0.000000	0.000427	0.000000
00 − 04/05	−0.007833	−0.263583	0	0	246	0.000000	0.007258	0.000000	0.000178	−0.000178	−0.000178
01 − 05/06	−0.129261	−0.879665	2	50	196	0.009229	0.009066	0.002763	0.000773	0.001990	0.001586
02 − 06/07	−0.247342	−1.085736	0	0	244	0.000000	0.016580	0.000000	0.001223	−0.001223	−0.001223

Table 13.14a: 5-years training and 1-year out-of-sample Hang Seng results. "Sample" denotes the training period used followed by the out-of-sample testing period (for example 97-99 implies that training data from 1997, 1998 and 1999 have been used to derive a trading rule which is then applied out-of-sample to data from 2000 and so on). "Excess" measures the fitness implied by a GP trading rule defined as excess return over a buy-and-hold strategy during the out-of-sample period, i.e. $(r_{gp} - r_{bh})$. ΔSOR indicates the excess Sortino ratio defined as $(SOR_{gp} - SOR_{bh})$ (both annualized) over the specified out-of-sample period. $\#T$ indicates the number of trades executed by a trading rule during out-of-sample testing with N_b and N_s denoting the number of buy-days (in-the-market) and sell-days (out-of-the-market), respectively. σ_b and σ_s indicate the standard deviation of returns during GP-in-market-days and GP-out-of-market-days, respectively. $\bar{r}_b$ and $\bar{r}_s$ denote the mean daily market return during GP-in-days and GP-out-days with $(\bar{r}_b - \bar{r}_s)$ as the difference between the two. $(\bar{r}_b - \bar{r}_m)$ measures the difference between mean daily returns during GP-in-days and buy-and-hold.

Sample	r_{gp}	r_{bh}	Δr	SOR_{gp}	SOR_{bh}	ΔSOR
Panel A: Transaction costs 0.1%						
97 − 01/02	−0.230937	−0.198992	−0.031945	−1.241062	−1.085768	−0.155294
98 − 02/03	0.120248	0.292751	−0.172503	0.649639	1.916353	−1.266714
99 − 03/04	−0.099843	0.103801	−0.203644	−0.536077	0.660709	−1.196786
00 − 04/05	−0.029741	0.041904	−0.071645	−0.233726	0.331433	−0.565159
01 − 05/06	0.039625	0.287605	−0.247980	0.000000	1.887073	−1.887073
02 − 06/07	0.041006	0.296347	−0.255342	0.000000	1.118694	−1.118694
Panel B: Transaction costs 0.25%						
97 − 01/02	−0.129163	−0.201992	0.072829	−0.699276	−1.102137	0.402861
98 − 02/03	0.122414	0.289751	−0.167336	0.658014	1.896715	−1.238700
99 − 03/04	0.100801	0.100801	0.000000	0.641614	0.641614	0.000000
00 − 04/05	0.092740	0.038904	0.053836	0.749980	0.305865	0.444115
01 − 05/06	0.178724	0.284605	−0.105882	1.252251	1.866321	−0.614069
02 − 06/07	0.041006	0.293347	−0.252342	0.000000	1.106381	−1.106381
Panel C: Transaction costs 0.5%						
97 − 01/02	−0.265044	−0.206992	−0.058052	−1.427454	−1.129419	−0.298035
98 − 02/03	0.208566	0.284751	−0.076184	1.230434	1.871636	−0.641202
99 − 03/04	0.095801	0.095801	0.000000	0.612220	0.612220	0.000000
00 − 04/05	0.026071	0.033904	−0.007833	0.000000	0.263583	−0.263583
01 − 05/06	0.150344	0.279605	−0.129261	0.951149	1.830815	−0.879665
02 − 06/07	0.041006	0.288347	−0.247342	0.000000	1.085736	−1.085736

Table 13.14b: 5-years training and 1-year out-of-sample Hang Seng results. “Sample” denotes the length of training and subsequent out-of-sample period. r_{gp} and r_{bh} denote the annualized out-of-sample returns for the GP trading rule and buy-and-hold, respectively. Δr is the difference between them. SOR_{gp} and SOR_{bh} indicate the repective annualized Sortino ratios for the GP trading rule and buy-and-hold, ΔSOR measures the difference between the two and is equal to ΔSOR in the preceding table.

Sample	Excess	ΔSOR	$\#T$	N_b	N_s	σ_b	σ_s	$\bar{r}_b$	$\bar{r}_s$	$(\bar{r}_b - \bar{r}_s)$	$(\bar{r}_b - \bar{r}_m)$
Panel A: Transaction costs 0.1%											
97 – 01/02 – 03	−0.182229	−0.524252	16	219	275	0.011495	0.011497	−0.000294	0.000607	−0.000901	−0.000502
98 – 02/03 – 04	−0.342232	−1.139331	5	218	278	0.011192	0.009916	0.000336	0.001241	−0.000905	−0.000507
99 – 03/04 – 05	−0.269518	−0.897080	13	234	261	0.009133	0.008617	−0.000474	0.001001	−0.001475	−0.000778
00 – 04/05 – 06	−0.155689	−0.565335	14	333	160	0.008253	0.008233	0.000562	0.000944	−0.000382	−0.000124
01 – 05/06 – 07	−0.545040	−1.243238	1	59	432	0.012718	0.013444	−0.000191	0.001427	−0.001618	−0.001424
Panel B: Transaction costs 0.25%											
97 – 01/02 – 03	−0.030089	−0.097582	4	240	254	0.011799	0.011218	0.000305	0.000115	0.000189	0.000097
98 – 02/03 – 04	−0.064590	−0.189059	2	294	202	0.010388	0.010653	0.001191	0.000337	0.000854	0.000348
99 – 03/04 – 05	0.001167	0.001511	7	374	121	0.009239	0.007712	0.000456	−0.000169	0.000626	0.000153
00 – 04/05 – 06	0.053836	0.173370	1	399	94	0.008461	0.007146	0.000969	−0.000517	0.001486	0.000283
01 – 05/06 – 07	−0.105882	−0.359497	3	322	169	0.015118	0.009124	0.001495	0.000732	0.000764	0.000263
Panel C: Transaction costs 0.5%											
97 – 01/02 – 03	−0.055338	−0.164355	4	375	119	0.011274	0.012206	0.000187	0.000272	−0.000085	−0.000021
98 – 02/03 – 04	−0.243735	−0.841709	5	279	217	0.010973	0.009870	0.000736	0.000982	−0.000246	−0.000108
99 – 03/04 – 05	−0.016668	−0.069520	4	387	108	0.009334	0.007078	0.000400	−0.000043	0.000443	0.000097
00 – 04/05 – 06	−0.262580	−1.168158	0	0	493	0.000000	0.008240	0.000000	0.000686	−0.000686	−0.000686
01 – 05/06 – 07	−0.129261	−0.440581	2	295	196	0.015575	0.009066	0.001538	0.000773	0.000765	0.000305

Table 13.15a: 5-years training and 2-years out-of-sample Hang Seng results. "Sample" denotes the training period used followed by the out-of-sample testing period (for example 97-99 implies that training data from 1997, 1998 and 1999 have been used to derive a trading rule which is then applied out-of-sample to data from 2000 and so on)."Excess" measures the fitness implied by a GP trading rule defined as excess return over a buy-and-hold strategy during the out-of-sample period, i.e. $(r_{gp} - r_{bh})$. ΔSOR indicates the excess Sortino ratio defined as $(SOR_{gp} - SOR_{bh})$ (both annualized) over the specified out-of-sample period. $\#T$ indicates the number of trades executed by a trading rule during out-of-sample testing with N_b and N_s denoting the number of buy-days (in-the-market) and sell-days (out-of-the-market), respectively. σ_b and σ_s indicate the standard deviation of returns during GP-in-market-days and GP-out-of-market-days, respectively. $\bar{r}_b$ and $\bar{r}_s$ denote the mean daily market return during GP-in-days and GP-out-days with $(\bar{r}_b - \bar{r}_s)$ as the difference between the two. $(\bar{r}_b - \bar{r}_m)$ measures the difference between mean daily returns during GP-in-days and buy-and-hold.

Sample	r_{gp}	r_{bh}	Δr	SOR_{gp}	SOR_{bh}	ΔSOR
Panel A: Transaction costs 0.1%						
97 − 01/02 − 03	−0.040868	0.050246	−0.091115	−0.228732	0.295519	−0.524252
98 − 02/03 − 04	0.037048	0.208164	−0.171116	0.201121	1.340452	−1.139331
99 − 03/04 − 05	−0.060650	0.074109	−0.134759	−0.381473	0.515606	−0.897080
00 − 04/05 − 06	0.090202	0.168047	−0.077845	0.637489	1.202824	−0.565335
01 − 05/06 − 07	0.029034	0.301554	−0.272520	0.131546	1.374784	−1.243238
Panel B: Transaction costs 0.25%						
97 − 01/02 − 03	0.033702	0.048746	−0.015044	0.189115	0.286697	−0.097582
98 − 02/03 − 04	0.174369	0.206664	−0.032295	1.141734	1.330793	−0.189059
99 − 03/04 − 05	0.073192	0.072609	0.000584	0.506171	0.504660	0.001511
00 − 04/05 − 06	0.193464	0.166546	0.026918	1.363417	1.190046	0.173370
01 − 05/06 − 07	0.247113	0.300054	−0.052941	1.007509	1.367006	−0.359497
Panel C: Transaction costs 0.5%						
97 − 01/02 − 03	0.018577	0.046246	−0.027669	0.108166	0.272521	−0.164355
98 − 02/03 − 04	0.082296	0.204164	−0.121868	0.475662	1.317372	−0.841709
99 − 03/04 − 05	0.061775	0.070109	−0.008334	0.416819	0.486340	−0.069520
00 − 04/05 − 06	0.032756	0.164046	−0.131290	0.000000	1.168158	−1.168158
01 − 05/06 − 07	0.232923	0.297554	−0.064631	0.913318	1.353899	−0.440581

Table 13.15b: 5-years training and 2-years out-of-sample Hang Seng results. “Sample” denotes the length of training and subsequent out-of-sample period. r_{gp} and r_{bh} denote the annualized out-of-sample returns for the GP trading rule and buy-and-hold, respectively. Δr is the difference between them. SOR_{gp} and SOR_{bh} indicate the repective annualized Sortino ratios for the GP trading rule and buy-and-hold, ΔSOR measures the difference between the two and is equal to ΔSOR in the preceding table.

Sample	Excess	ΔSOR	$\#T$	N_b	N_s	σ_b	σ_s	$\bar{r}_b$	$\bar{r}_s$	$(\bar{r}_b - \bar{r}_s)$	$(\bar{r}_b - \bar{r}_m)$
Panel A: Transaction costs 0.1%											
997 − 01/02 − 04	−0.303167	−0.596732	16	219	524	0.011495	0.010941	−0.000294	0.000554	−0.000849	−0.000598
98 − 02/03 − 05	−0.344164	−0.801664	7	430	313	0.009443	0.009661	0.000271	0.001106	−0.000835	−0.000352
99 − 03/04 − 06	−0.400346	−0.916821	18	430	312	0.009015	0.008861	0.000125	0.001251	−0.001126	−0.000473
00 − 04/05 − 07	−0.298276	−0.548746	16	515	223	0.012134	0.010564	0.000687	0.001345	−0.000659	−0.000199
Panel B: Transaction costs 0.25%											
97 − 01/02 − 04	0.011665	0.022471	5	409	334	0.011095	0.011127	0.000595	−0.000051	0.000646	0.000290
98 − 02/03 − 05	−0.110072	−0.244786	2	401	342	0.009592	0.009483	0.000821	0.000390	0.000430	0.000198
99 − 03/04 − 06	−0.185420	−0.426435	18	513	229	0.009270	0.008249	0.000616	0.000561	0.000055	0.000017
00 − 04/05 − 07	0.053836	0.043995	1	644	94	0.012191	0.007146	0.001090	−0.000517	0.001607	0.000205
Panel C: Transaction costs 0.5%											
97 − 01/02 − 04	−0.124409	−0.250985	5	463	280	0.010954	0.011373	0.000286	0.000334	−0.000048	−0.000018
98 − 02/03 − 05	−0.343745	−0.799090	9	443	300	0.009824	0.009106	0.000411	0.000936	−0.000526	−0.000212
99 − 03/04 − 06	−0.233037	−0.550238	12	551	191	0.009408	0.007547	0.000543	0.000759	−0.000216	−0.000056
00 − 04/05 − 07	−0.537523	−1.107763	0	0	738	0.000000	0.011679	0.000000	0.000886	−0.000886	−0.000886

Table 13.16a: 5-years training and 3-years out-of-sample Hang Seng results. "Sample" denotes the training period used followed by the out-of-sample testing period (for example 97-99 implies that training data from 1997, 1998 and 1999 have been used to derive a trading rule which is then applied out-of-sample to data from 2000 and so on). "Excess" measures the fitness implied by a GP trading rule defined as excess return over a buy-and-hold strategy during the out-of-sample period, i.e. $(r_{gp} - r_{bh})$. ΔSOR indicates the excess Sortino ratio defined as $(SOR_{gp} - SOR_{bh})$ (both annualized) over the specified out-of-sample period. $\#T$ indicates the number of trades executed by a trading rule during out-of-sample testing with N_b and N_s denoting the number of buy-days (in-the-market) and sell-days (out-of-the-market), respectively. σ_b and σ_s indicate the standard deviation of returns during GP-in-market-days and GP-out-of-market-days, respectively. $\bar{r}_b$ and $\bar{r}_s$ denote the mean daily market return during GP-in-days and GP-out-days with $(\bar{r}_b - \bar{r}_s)$ as the difference between the two. $(\bar{r}_b - \bar{r}_m)$ measures the difference between mean daily returns during GP-in-days and buy-and-hold.

Sample	r_{gp}	r_{bh}	Δr	SOR_{gp}	SOR_{bh}	ΔSOR
Panel A: Transaction costs 0.1%						
97 − 01/02 − 04	−0.026366	0.074690	−0.101056	−0.147168	0.449565	−0.596732
98 − 02/03 − 05	0.038860	0.153581	−0.114721	0.242589	1.044253	−0.801664
99 − 03/04 − 06	0.014020	0.147469	−0.133449	0.089631	1.006452	−0.916821
00 − 04/05 − 07	0.117773	0.217199	−0.099425	0.576192	1.124938	−0.548746
Panel B: Transaction costs 0.25%						
97 − 01/02 − 04	0.077578	0.073690	0.003888	0.466017	0.443545	0.022471
98 − 02/03 − 05	0.115890	0.152581	−0.036691	0.792007	1.036792	−0.244786
99 − 03/04 − 06	0.084662	0.146469	−0.061807	0.573005	0.999439	−0.426435
00 − 04/05 − 07	0.234144	0.216199	0.017945	1.162566	1.118571	0.043995
Panel C: Transaction costs 0.5%						
97 − 01/02 − 04	0.030554	0.072023	−0.041470	0.183099	0.434083	−0.250985
98 − 02/03 − 05	0.036333	0.150914	−0.114582	0.225120	1.024210	−0.799090
99 − 03/04 − 06	0.067123	0.144802	−0.077679	0.437346	0.987584	−0.550238
00 − 04/05 − 07	0.035358	0.214532	−0.179174	0.000000	1.107763	−1.107763

Table 13.16b: 5-years training and 3-years out-of-sample Hang Seng results. “Sample” denotes the length of training and subsequent out-of-sample period. r_{gp} and r_{bh} denote the annualized out-of-sample returns for the GP trading rule and buy-and-hold, respectively. Δr is the difference between them. SOR_{gp} and SOR_{bh} indicate the repective annualized Sortino ratios for the GP trading rule and buy-and-hold, ΔSOR measures the difference between the two and is equal to ΔSOR in the preceding table.

	Trading Rule	*1-year out-of-sample* Excess	ΔSOR	*2-years out-of-sample* Excess	ΔSOR	*3-years out-of-sample* Excess	ΔSOR
3-years in-sample	$97-99/...c025$	0.202731	0.446743	0.067099	0.146903	0.088498	0.097996
	$98-00/...c025$	0.255669	0.935284	0.248648	0.542211	0.184907	0.291989
	$97-99/...c05$			0.301239	0.523493	0.277910	0.360544
	$98-00/...c05$	0.110108	0.590780	0.058290	0.194659		
	$99-01/...c05$	0.063992	0.378851				
5-years in-sample	$97-01/...c025$	0.072829	0.402861			0.011665	0.022471
	$99-03/...c025$			0.001167	0.001511		
	$00-04/...c025$	0.053836	0.444115	0.053836	0.173370	0.053836	0.043995

Table 13.17: Best Genetic Programming trading rules for the Hang Seng.

13.4 Conclusions about Market Efficiency in the DAX and the Hang Seng

Jensen (1978) considered markets to be efficient if there are no gains from trading based on available information after risk adjustment and transaction costs. In the thesis at hand, GP was tasked to find trading rules based on the most common set of information, i. e. closing prices[106] therefore testing for weak market efficiency. Appropriate transaction costs ($c = 0.25, 0.5$) were included in the analysis and the Sortino ratio acted as a measure for risk-adjusted returns. The results of the GP-optimized trading rules in terms of market efficiency indicate that in general, the EMH apparently holds for both the DAX and the Hang Seng for the combined out-of-sample period as the analysis of long term GP performance[107] at best showed marginal (DAX) and zero (Hang Seng) risk-adjusted excess returns after transaction costs.

However, the picture is not so clear on a smaller level. A couple of rules outperformed the benchmark in both indices. What is even more interesting is that one rule in the Hang Seng managed to do so in an overall rising market[108]. For realistic transaction costs, GP beats the benchmark on a risk-adjusted basis in 13 out of 72 scenarios for the DAX and 17 out of 72 scenarios for the Hang Seng. In addition, some of the successful rules have a surprisingly easy structure[109]. It is debatable whether any technical analyst would have coincidentally come up with the rules suggested by GP as his/her human brainchild, however one has to bear in mind that the concept of GP was well-known and available already at the beginning of the data sample in the year 2000. It must also be emphasized that the rules, due to the data division scheme and rolling time-frame approach, are so-to-say ex-ante optimal, i. e. GP as a machine learning tool evolved trading rules from training samples that were then applied blindly to data unknown to the algorithm beforehand and GP still managed to beat the benchmark in a considerable amount of cases. Some of these rules, particularly those during the 2002 out-of-sample year in the DAX, manage to beat the benchmark by

[106] Which are available to anybody at a very low or even no cost at all.
[107] The scenarios with updated trading rules at each year's beginning.
[108] The 00-04/...c025 rule managed to outperform the benchmark on a risk-adjusted basis.
[109] See Figures 13.3 and 13.6.

simply switching out-of-the-market and staying there thus saving the investor from tremendous losses in the index. This also applies to the Hang Seng for the years 2000-2002 as most of the successful rules come from that bearish market period. Critics might argue that this outperformance is not very convincing as a human investor would also have switched out-of-the-market during a market meltdown. However, apart from the fact that the choice of straight buy-and-hold as benchmark come hell or high water is simply a design decision for the study at hand[110], a human investor does not ex-ante know whether the market is already in a meltdown neither does he know how long it will take the market to bottom out and how much losses will still be incurred (neither does GP) and when to re-enter the market. In contrast, GP, while it also cannot tell the future, comes up with a quantifiable strategy for this period which simply suggests to stay out-of-the-market for a prolonged period of time and eventually go long again. This is what GP independently recommends in both markets so it it highly unlikely that these rules simply emerged by chance.

Why is it then that GP (almost) fails in the long-run? Chen and Navet (2006, 2007) investigate this issue and point out the distinction between efficient market and inefficent algorithms, so the question is which of the two is the reason for this finding? It might be that markets have indeed been simply efficient for the period 2000-2007 used in this study, however things would look differently if one would just take say the 2000-2003 sample. In this case, particularly the DAX would look all but efficient. Chen et al. (2008) emphasize the link between market conditions during traing and out-of-sample period. They argue that in a steadily rising market, the best possible outcome for GP often is to simply replicate buy-and-hold. There may be better strategies but finding them might be very hard for GP - provided they exist at all. Both DAX and Hang Seng feature sustained upward trends from the year 2003-2007 which tipped the scales in favour of buy-and-hold. Therefore, the failure of GP in the long run may be partly due to the design decision to take a rather recent data sample. One of the consequences for future studies might be to focus on a market that has shown a rather trendless/mean reverting pattern during the last couple of

[110] Buy-and-hold by design favours GP during bearish markets while the opposite is true for bullish markets.

years[111].

Another issue might be inherent deficiencies in the GP algorithm. As already pointed out, the issue of parameter settings within GP is particularly important and unfortunately, as pointed out in Chen et al. (2008), is not well understood. In addition, the choice of a suitable function set for GP is still in open issue (Navet and Chen, 2007). Therefore, the question as to whether the results might be improved by using different sets of parameters and function sets is still open as well.

[111] Further directions for future research will be pointed out in the next chapter.

Part V

Summary and Conclusion

14 Genetic Programming and Market Efficiency

The thesis started with a literature review followed by a technical chapter which presented the inner workings of GP. The next chapter dealt with the application of GP-optimized trading rules to the DAX and the Hang Seng. The results indicated that GP performance depends on market cycles; it manages to beat the buy-and-hold benchmark even after risk adjustment and transaction costs in bear markets whereas, in the case of the DAX, GP at best marginally outperforms the benchmark during a bullish market cycle as seen in the years 2003-2007. Things look slightly different for the Hang Seng as GP in one case even manages to beat the benchmark by a considerable margin during an overall upward trend in the market. Therefore, these results imply that the EMH does not always hold. However, the results from applying a yearly updated GP trading rule to the whole data sample showed only a very marginal risk-adjusted outperformance in the DAX and no outperformance in the Hang Seng. Nevertheless it must be emphasized that GP questions the absolute validity of the EMH as one might have expected that GP would generally fail at beating the respective benchmarks. This has been proven wrong given the amount of instances where GP outperforms the benchmark and at times even does so when the market apparently seems to be efficient by exhibiting a strong upward trend. Therefore, though GP maybe failed to deliver the ultimate proof that markets are inefficient, it casts at least some doubts on the validity of the EMH and implies that the EMH does not always hold.

15 Directions for Future Research

During the writing of the thesis at hand, a couple of ideas for further GP research came to the author's mind which justify a thesis on their own. The author hopes that one of these issues will eventually be picked up to continue this line of research. Referring to the literature review in the second chapter, a typical GP application for testing market efficiency consists of:

1. *Choice of a particular market/asset class*
2. *Choice of input variables, data sample and data division scheme*
3. *Parameter settings, choice of function set and selection algorithm*
4. *Choice of fitness function and benchmark strategy.*

These items provide endless variation of GP-related research. Based on the the literature review and the experience gained as part of the study at hand, directions for future research can be narrowed down to a more reasonable and promising perspective that may assist in a better understanding of financial markets and the mechanics of GP.

From a personal point of view, the existing literature might be extended by investigating GP performance using macroeconomic variables. So far only Bauer (1994) and Ammann and Zenkner (2003) use fundamental variables such as inflation, growth and interest rate spreads as input and report encouraging results. Major sources of inspiration for macroeconomic GP inputs also come from the existing literature on modeling excess returns. Promising candidates for macroeconomic GP inputs are (in no particular order):

- dividend yields

  (Fama and French, 1989; Chen, 1991; Bekaert and Hodrick, 1992; Pesaran and Timmermann, 1995; Olson and Mossmann, 2001; Döpke et al., 2008)

- T-bill rates from 1-12 months

  (Breen et al., 1989; Chen, 1991; Pesaran and Timmermann, 1995; Olson and Mossmann, 2001; Döpke et al., 2008)

- term spread

  (Keim and Stambaugh, 1986; Chen, 1991; Bekaert and Hodrick, 1992; Olson and Mossmann, 2001; Fama and French, 1989, Döpke et al., 2008)

- default spread

  (Fama and French, 1989; Chen, 1991; Olson and Mossmann, 2001)

- inflation

  (Pesaran and Timmermann, 1995; Olson and Mossmann, 2001)

- changes in industrial production/GNP growth

  (Chen, 1991; Pesaran and Timmermann, 1995; Olson and Mossmann, 2001).

Even though the prospect of finding evidence against the EMH may be limited, the use of macroeconomic variables may yield further insight into the relationship between stock markets and the overall macroeconomic environment and may point out major structural breaks in stock markets. Another promising avenue for GP applied to stock markets may be to look for possible inefficiencies on a much smaller scale using high-frequency price data. Seasoned market participants may argue that ineffiencies build up and exist just for a couple of seconds. Therefore, using a finer search grid may yield results that contradict the EMH[112]. However, this would also require way more data and is technically more demanding in terms of CPU time.

Concerning other asset classes, the GP literature on FOREX markets does not seem to offer much space for a further contribution. As currencies can be basically traded anywhere in the world, the distinction between different marketplaces such as for stocks does not exist. The author feels that the various papers by Neely et al. adequately cover the topic and cannot be much improved upon except for using a more up-to-date data sample. More surprisingly, it turned out that the bond market has almost not been covered at all by the existing literature. The only approach the author is aware of is Bauer (1994), who uses a classic GA to switch between long- and short-term government bonds on the one hand and between risky corporate bond and safe-haven government bonds on the other hand. Though the reported results are mixed, the lack of research

[112] All existing studies including this study use daily and sometimes even monthly data. From a personal point of view, using high-frequency data is the most promising avenue of research.

on this issue opens up avenues for further experiments. It would be worthwhile to re-investigate the bond market application proposed by Bauer (1994) for European corporate bonds/government bonds using a GP setup with appropriate inputs such as growth and inflation expectations, price momentum, value of equities vs. bonds and flows into mutual bond funds.

A gap exists in the literature as well when it comes to commodity markets. However, commoditiy futures might be difficult to implement in GP since this market is plagued probably more than others by (extremely) high volatility, many speculative traders and highly disrupted trading patterns. This might explain the lack of research directed at this market. Regarding other markets, it would also be worthwhile to investigate derivative markets, especially the market for credit derivatives.

The impact of parameter choice within a GP framework is not well understood (Chen et al., 2008). Depending on parameter settings, the search space may not be covered thoroughly. It might be interesting to test several parameter sets to better understand how GP discovers trading rules in a market. It might prove beneficial to conduct this kind of comparative statics on an artificial time series first and then try to match this time series with real-world data. As another avenue for future research, several studies advocate tinkering with the fitness function rather than with the underyling parameters such as crossover and mutation rates to fine-tune results (Amman and Zenkner, 2003; Becker and Seshadri, 2003b; Navet and Chen, 2007). The fitness function is at the core of evolutionary modeling of financial markets. While the usual approach is to simply measure excess returns, more sophisticated fitness functions could be employed. For example, volatility could be included as well in a (appropriately weighted) fitness function as well as a complexity-penalizing factor that kicks in once too many trades (and therefore too much transaction costs) would be carried out during the training period as shown in Becker and Seshadri (2003b). Variations thereof, including appropriate weighting schemes, could be another avenue for future resarch. Navet and Chen (2007) present a wealth of further ideas on how the literature on GP can be improved upon. Most of the issues addressed in their paper are still an open question at the time of this writing (March 2010).

References

[1] **Alexander, S.** (1961). Price Movements in Speculative Markets: Trends or Random Walks. *Industrial Management Review* 2, 7-26.

[2] **Allen, F.** and Karjalainen, R. (1999). Using Genetic Algorithms to Find Technical Trading Rules. *Journal of Financial Economics* 51, 245-271.

[3] **Altenberg, L.** (1994). The evolution of evolvability in genetic programming. In: Kinnear, K.E. (ed.), Advances in Genetic Programming, MIT Press.

[4] **Ammann, M.** and Zenkner, C. (2003). Tactical Asset Allocation mit Genetischen Algorithmen. *Schweizerische Zeitschrift für Volkswirtschaft und Statistik* 139(1), 1-40.

[5] **Austin, M. P.**, Bates, G., Dempster, M., Leemans, V., Williams, S. N. (2004). Adaptive systems for foreign exchange trading. *Quantitative Finance* 4(4), 37-45.

[6] **Azzini, A.** and Tettamanzi, A.G.B. (2008). Evolutionary Single-Position Automated Trading. In: Giacobini, M. et al. (eds.), Proceedings of the European Workshop on Evolutionary Computation in Finance and Economics, EVOFIN 08, Springer-Verlag.

[7] **Babcock, B.** (1989). The Dow Jones-Irwin Guide to Trading Systems, Dow Jones-Irwin.

[8] **Bachelier, L.** (1900). Théorie de la Spéculation, Annales Scientifiques de l'Ecole Normale Superieure, III-17, 21-86.

[9] **Banzhaf, W.**, Nordin, P., Keller, R. E. and Francone, F. D. (1998). Genetic Programming-An Introduction: On the Automatic Evolution of Computer Programs and Its Applications, Morgan Kaufmann Publishers.

[10] **Basu, S.** (1977). Investment Performance of Common Stocks in Relation to Their Price-Earnings Ratios: A Test of the Efficient Market Hypothesis. *Journal of Finance* 32(3), 663-682.

[11] **Bauer, R. J.** and Liepins, G. E. (1992). Genetic Algorithms and Computerized Trading Strategies. In: O'Leary, D. E., Watkins, R. R. (eds.), Expert Systems in Finance, North Holland.

[12] **Bauer, R. J.** (1994). Genetic Algorithms and Investment Strategies, John Wiley & Sons.

[13] **Becker, L. A.** and Seshadri, M. (2003a). GP-evolved technical trading rules can outperform buy and hold. In: Proceedings of the Sixth International Conference on Computational Intelligence and Natural Computing, Embassy Suites Hotel and Conference Center, Cary, North Carolina.

[14] **Becker, L. A.** and Seshadri, M. (2003b). Comprehensibility and overfitting avoidance in genetic programming for technical trading rules. Technical Report, Worcester Polytechnic Institute.

[15] **Bekaert, G.** and Hodrick, R. J. (1992). Characterizing Predictable Components in Excess Returns on Equity and Foreign Exchange Markets. *Journal of Finance* 47(2), 467-509.

[16] **Bessembinder, H.** and Chan, K. (1995). The Profitability of Technical Trading Rules in the Asian Stock Markets. *Pacific Basin Finance Journal* 3, 257-284.

[17] **Bessembinder, H.** and Chan, K. (1998). Market Efficiency and the Returns to Technical Analysis. *Financal Management* 27, 5-17.

[18] **Bhattacharyya, S.** and Mehta, K. (2002). Evolutionary Induction of Trading Models. In: Chen, S.-H. (ed.), Evolutionary Computation in Economics and Finance, Physica-Verlag.

[19] **Bollerslev, T.** (1986). Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics* 31, 307-327.

[20] **Bollerslev, T.** and Hodrick, R. J. (1994). Financial Market Efficiency Tests. In: Pesaram, H. and Wickens, M. (eds.), Handbook of Applied Econometrics Vol. I, Basil Blackwell.

[21] **Breen, W.**, Glosten, L. R. and Jagannathan, R. (1989). Economic Significance of Predictable Variations in Stock Index Returns. *Journal of Finance* 44(5), 1177-1189.

[22] **Brock, W. A.**, Dechert, W. and Scheinkman, J. (1987). A Test for Independence based on the Correlation Dimension. University of Wisconsin, Madison, University of Houston and University of Chicago.

[23] **Brock, W. A.**, Hsieh, D. A. and LeBaron, B. (1991). Nonlinear Dynamics, Chaos and Instability: Statistical Theory and Economic Evidence, MIT Press.

[24] **Brock, W. A.**, Lakonishok, J. and LeBaron, B. (1992). Simple Technical Trading Rules and the Stochastic Properties of Stock Returns. *Journal of Finance* 47, 1731-1764.

[25] **Brown, S.**, Goetzmann, W. and Kumar, A. (1998). The Dow Theory: William Peter Hamilton's Track Record Reconsidered. *Journal of Finance* 53(4), 1311-1333.

[26] **Cantú-Paz, E.** et al. (eds.) (2003). Genetic and Evolutionary Computation - GECCO 2003: Genetic and Evolutionary Computation Conference, Chicago, IL, Proceedings: Pt. 2, Springer-Verlag.

[27] **Chaudry, A.** and Johnson, H. L. (2008). The Efficacy of the Sortino Ratio and Other Benchmarked Performance Measures Under Skewed Return Distributions. *Australian Journal of Management* 32, 485-502.

[28] **Chen, N.** (1991). Financial Investment Opportunities and the Macroeconomy. *Journal of Finance* 46(2), 529-554.

[29] **Chen, S.-H.** and Yeh, C.-H. (1996). Genetic Programming and the Efficient Market Hypothesis. In: Genetic Programming 1996: Proceedings of the First Annual Conference, Stanford University, CA, USA, MIT Press.

[30] **Chen, S.-H.** and Yeh, C.-H. (1997). Toward a computable approach to the efficient market hypothesis: An application of genetic programming. *Journal of Economic Dynamics and Control* 21, 1043-1063.

[31] **Chen, S.-H.** and Kuo, T.-W. (2001). Trading strategies on trial: A comprehensive review of 21 practical trading strategies over 56 listed stocks. In: Proceedings of the Fourth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2001), 66-71, IEEE Computer Society Press.

[32] **Chen, S.-H.** (ed.) (2002a). Evolutionary Computation in Economics and Finance, Physica-Verlag.

[33] **Chen, S.-H.** (ed.) (2002b). Genetic Algorithms and Genetic Programming in Computational Finance, Kluwer Academic Publishers.

[34] **Chen, S.-H.**, Kuo, T.-W. and Y.-P. Shieh (2002). Genetic programming: A tutorial with the software simple GP. In: Chen, S.-H. (ed.), Genetic Algorithms and Genetic Programming in Computational Finance, 55-77, Kluwer Academic Publishers.

[35] **Chen, S.-H.** and Kuo, T.-W. (2003a). Overfitting or poor learning: A critique of current financial applications of GP. In: Ryan, C., Soule, T., Keijzer, M., Tsang, E., Poli, R. and Costa, E. (eds.), Proceedings of the Sixth European Conference on Genetic Programming (EuroGP-2003), Essex, Lecture Notes in Computer Science 2610, 24-46, Springer-Verlag.

[36] **Chen, S.-H.** and Kuo, T.-W. (2003b). Discovering hidden patterns with genetic programming. In: Chen, S.-H and Wang, P. P. (eds.), Computational Intelligence in Economics and Finance, 329-347, Springer-Verlag.

[37] **Chen, S.-H.** and Navet, N. (2006). Pretests for genetic programming evolved trading programs: zero-intelligence strategies and lottery trading. In: King, I., Wang, J., Chan, L. and Wang, D.L. (eds.), Neural Information Processing, 13th International Conference ICONIP 2006 Hong Kong, Proceedings, Part III, Lecture Notes in Computer Science, 450-460, Springer-Verlag.

[38] **Chen, S.-H.** and Navet, N. (2007). Failure of Genetic-Programming Induced Trading Strategies: Distinguishing between Efficient Markets and Inefficient Algorithms. In: Chen, S.-H., Wang, P. P. and Kuo, T.-W.

(eds.), Computational Intelligence in Economics and Finance Vol. II., Springer-Verlag.

[39] **Chen, S.-H.**, Kuo, T.-W. and Hoi, K.-M. (2008). Genetic Programming and Financial Trading: How Much About "What We Know". In: Zopounidis, C., Doumpos, M. and Pardalos, P. M. (eds.), Handbook of Financial Engineering, Springer-Verlag.

[40] **Colin, A. M.** (1994). Genetic Algorithms for Financial Modeling. In: Deboeck, G. J. (ed.), Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets, John Wiley & Sons.

[41] **Colin, A. M.** (2000). A genetic programming-based approach to generation of foreign exchange trading models. In: Barnett, W. A. et al. (eds.), Commerce, Complexity, and Evolution. Proceedings of the Twelfth International Symposium in Economic Theory and Econometrics, Cambridge University Press.

[42] **Costa Pereira, C. da** and Tettamanzi, A. G. B. (2006). Fuzzy-Evolutionary Modeling for Single-Position Day Trading. In: Brabazon, A. and O'Neill, M. (eds.), Natural Computing in Computational Finance, Studies in Computational Intelligence 100, Springer-Verlag.

[43] **Davis, L.** (1989). Adapting operator probabilities in genetic algorithms. In: Schaffer, J.D. (ed.), Proceedings of the Third International Conference on Genetic Algorithms 61-69, Morgan Kaufmann.

[44] **Deboeck, G. J.** (1994). Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets, John Wiley & Sons.

[45] **Dempster, M.** and Jones, C. (2001). A real-time adaptive trading system using genetic programming. *Quantitative Finance* 1, 397-413.

[46] **Döpke, J.**, Hartmann, D. and Pierdzioch, C. (2008). Real-time macroeconomic data and ex ante stock return predictability. *International Review of Financial Analysis* 17, 274-290.

[47] **Drezewski, R.** and Sepielak, J. (2008). Evolutionary System for Generating Investment Strategies. In: Giacobini, M. et al. (eds.), Proceedings

of the European Workshop on Evolutionary Computation in Finance and Economics, EVOFIN 08, Springer-Verlag.

[48] **Eiben, A. J.**, Smith, J. E. (2003). Introduction to Evolutionary Computing, Springer-Verlag.

[49] **Engle, R. F.** (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica* 50, 987-1007.

[50] **Fama, E. F.** (1965a). The Behavior of Stock-Market Prices. *The Journal of Business* 38(1), 34-105.

[51] **Fama, E. F.** (1965b). Random Walks in Stock Market Prices. *Financial Analysts Journal* 21(5), 5559.

[52] **Fama, E. F.** (1970). Efficient capital markets: A review of theory and empirical work. *Journal of Finance* 25, 383-417.

[53] **Fama, E. F.** and French, K. R. (1988). Dividend Yields and Expected Stock Returns. *Journal of Financial Economics* 22, 3-25.

[54] **Fama, E. F.** and French, K. R. (1989). Business Conditions and Expected Returns on Stocks and Bonds. *Journal of Financial Economics* 25, 23-49.

[55] **Fama, E. F.** (1991). Efficient capital markets: II. *Journal of Finance* 46, 1575-1617.

[56] **Fogel, L. J.**, Owens, A. J. and Walsh, M. J. (1965). Artificial Intelligence through a simulation of evolution. In: Callahan, A., Maxfield, M. and Fogel, L. J. (eds.), Biophysics and Cybernetic Systems. Spartan, Washington DC.

[57] **Fogel, L. J.**, Owens, A. J. and Walsh, M. J. (1966). Artificial Intelligence through Simulated Evolution, Wiley.

[58] **Franses, P. H.**, van Dijk, D. (2000). Non-Linear Time Series Models in Empirical Finance, Cambridge University Press.

[59] **French, K.** (1980). Stock Returns and the Weekend Effect. *Journal of Financial Economics* 5, 55-69.

[60] **Frick, A.**, Herrmann, R., Kreidler, M., Narr, A., Sesse, D. (1996). A Genetic-Based Approach for the Derivation of Trading Strategies on the German Stock Market. In: Amari, S., Xu, L., King, I., Leung, K. (eds.), Progress in Neural Information Processing, Vol. 2, Springer-Verlag.

[61] **Fyfe, C.**, Marney, J. P. and Tarbert, H. (1999). Technical analysis versus market efficiency - a genetic programming approach. *Applied Financial Economics* 9, 183-191.

[62] **Fyfe, C.**, Marney, J. P. and Tarbert, H. (2005). Risk Adjusted Returns from Technical Trading: a Genetic Programming Approach. *Applied Financial Economics* 15, 1073-1077.

[63] **Gang, W.** and Soule, T. (2004). How to choose appropriate function sets for GP. In: Keijzer, M., O'Reilly, U.M., Lucas, S.M., Costa, E. and Soule, T. (eds.), Genetic Programming 7th European Conference EuroGP 2004, Proceedings, Lecture Notes in Computer Science, 198-207, Springer-Verlag.

[64] **Goldberg, D.** (1989). Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley.

[65] **Granville, J. E.** (1976). New Strategy of Daily Stock Market Timing for Maximum Profit, Prentice-Hall.

[66] **Harland, Zac** (1999). Using Nonlinear Neurogenetic Models with Profit Related Objective Functions to Trade the US T-bond Future. In: Abu-Mostafa, Y. S., LeBaron, B. et al. (eds.), Computational Finance 1999, MIT Press.

[67] **Holland, J. J.** (1975). Adaption in Natural and Artificial Systems, University of Michigan Press.

[68] **Hsieh, D. A.** (1989). Testing for Nonlinear Dependence in Daily Foreign Exchange Rates. *Journal of Business* 62, 339-368.

[69] **Huang, Y. S.** (1995). The Trading Performance of Filter Rules On the Taiwan Sotck Exchange. *Applied Financial Economics* 5, 391-395.

[70] **Hudson, R.**, Dempsey, M. and Keasey, K. (1996). A Note on the Weak Form of Efficiency of Capital Markets: The Application of Simple Technical Trading Rules to the U.K. Stock Markets 1935-1994. *Journal of Banking and Finance* 20, 1121-1132.

[71] **Islam, S.M.N.** and Watanapalachaikul, S. (2005). Empirical Finance. Physica Series Contributions to Economics, Springer-Verlag.

[72] **Jaffe, J. F.** (1974). Special information and insider trading. *Journal of Business* 47, 410-428.

[73] **Jegadeesh, N.** and Titman, S. (1993). Returns to buying winners and selling losers: Implication for stock market efficiency. *Journal of Finance* 48, 65-91.

[74] **Jensen, Michael C.** (1978). Some anomalous evidence regarding market efficiency. *Journal of Financial Economics* 6(23), 95-101.

[75] **Kaboudan, M. A.** (2000). Genetic Programming Prediction of Stock Prices. *Computational Economics* 16(3), 207-236.

[76] **Kaboudan, M. A.** (2002). GP Forecasts of Stock Prices for Profitable Trading. In: S.-H. Chen (ed.), Evolutionary Computation in Economics and Finance, Physica-Verlag.

[77] **Karjalainen, R.** (2002). Evolving technical trading rules for S&P futures. In: Acar, E., Satchell, S. (eds.), Advanced Trading Rules, Butterworth-Heinemann.

[78] **Karr, C. L.**, Bowersox, R., Singh, V. (2003). Minimization of Sonic Boom on Supersonic Aircraft Using an Evolutionary Algorithm. In: Cantú-Paz, E. et al. (eds.), Genetic and Evolutionary Computation - GECCO 2003: Genetic and Evolutionary Computation Conference, Chicago, IL, Proceedings: Pt. 2, Springer-Verlag.

[79] **Kassicieh, S. K.**, Paez, T. L. and Vora, G. (1997). Investment decisions using genetic algorithms. In: Proceedings of the 30th Hawaii International Conference on System Sciences, IEEE Computer Society.

[80] **Keane, A.** (2001). An Introduction to Evolutionary Computing in Design Search and Optimization. In: Kallel, L., Naudts, B. and Rogers, A. (eds.), Theoretical Aspects of Evolutionary Computing, Springer-Verlag.

[81] **Keim, D. B.** and Stambaugh, R. F. (1986). Predicting Returns in the Stock and Bond Market. *Journal of Financial Economics* 17, 357-390.

[82] **Koza, J. R.** (1992). Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press.

[83] **Kwon, K.-Y.** and Kish, R.J. (2002). A comparative study of technical trading strategies and return predictability: an extension of Brock, Lakonishok, and LeBaron (1992) using NYSE and NASDAQ indices. *The Quarterly Review of Economics and Finance* 42, 611-631.

[84] **Kwon, Y.-K.**, Moon, B.-R. (2003). Daily Stock Prediction Using Neuro-genetic Hybrids. In: Cantú-Paz, E. et al. (eds.), Genetic and Evolutionary Computation - GECCO 2003: Genetic and Evolutionary Computation Conference, Chicago, IL, Proceedings: Pt. 2, Springer-Verlag.

[85] **Lam, S. S.**, Lam, K. P. and Ng, H. S. (2002). Genetic Fuzzy Expert Trading System For NASDAQ Stock Market Timing. In: Chen, S.-H. (ed.), Genetic Algorithms and Genetic Programming in Computational Finance, Kluwer Academic Publishers.

[86] **Langdon, W. B.** and Poli, R. (2002). Foundations of Genetic Programming, Springer-Verlag.

[87] **LeBaron, B.** (1999). Technical Trading Rule Profitability and Foregin Exchange Intervention. *Journal of International Economics* 49, 125-143.

[88] **Li, J.** and Tsang, P. K. (1999). Investment Decision Making Using FGP: A Case Study. In: Angeline, P. J. et al. (eds.), Proceedings of the Congress on Evolutionary Computation, Vol. 2, 1253-1259, IEEE Press.

[89] **Li, J.**, Tsang, P. K. (2002). EDDIE for financial forecasting. In: Chen, S.-H., (ed.), Genetic Algorithms and Genetic Programming in Computational Finance, 161-174, Kluwer Academic Press.

[90] **Lipinski, P.** (2007). ECGA vs. BOA in discovering stock market trading experts. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2007), 531-538, ACM Press.

[91] **Lo, A. W.**, MacKinlay, A. C. (1988). Stock Market Prices do not Follow Random Walks: Evidence from a Simple Specification Test. *Review of Financial Studies* 1(1), 41-66.

[92] **Lo, A. W.**, Mamaysky, H. and Wang, J. (2000). Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation. *Journal of Finance* 55, 1705-1765.

[93] **McNelis, P. D.** (2005). Neural Networks in Finance: Gaining Predicitive Edge in the Market, Elsevier Academic Press.

[94] **Medsker, L. R.** (1995). Hybrid Intelligent Systems, Kluwer Academic Publishers.

[95] **Miller, R. E.** and Reilly, F. K. (1987). An Examination of Mispricing, Returns and Uncertainty for Initial Public Offerings. *Financial Management* 16, 32-38.

[96] **Navet, N.** and Chen, S.-H. (2007). Financial Data Mining with Genetic Programming: A Survey and Look Forward. Invited paper at the 56th session of the International Statistical Institute (ISI 2007), Lisboa, Portugal. Downloadable at http://www.loria.fr/~nnavet/publi/ISI2007_NN_SHC_AT.pdf.

[97] **Navet, N.** and Chen, S.-H. (2008). On Predictability and Profitability. Would GP Induced Trading Rules be Sensitive to the Observed Entropy of Time Series? In: Brabazon, T. and O'Neill, M. (eds.), Natural Computing in Computational Finance, Springer-Verlag.

[98] **Nedjah, N.**, Abraham, A. and de Macedo Mourelle, L. (eds.) (2006). Genetic Systems Programming : Theory and Experiences, Springer-Verlag.

[99] **Neely, C.**, Weller, P. and Dittmar, R. (1997). Is Technical Analysis in the Foreign Exchange Market Profitable? A Genetic Programming Approach. *Journal of Financial and Quantitative Analysis* 32(4), 405-426.

[100] **Neely, C.** and Weller, P. A. (1999). Technical Trading Rules in the European Monetary System. *Journal of International Money and Finance* 18(3), 429-458.

[101] **Neely, C.** and Weller, P. A. (2001). Technical analysis and central bank intervention. *Journal of International Money and Finance* 20(7), 949-970.

[102] **Neely, C.** and Weller, P. A. (2003a). Intraday technical trading in the foreign exchange market. *Journal of International Money and Finance* 22(2), 223-237.

[103] **Neely, C.** (2003b). Risk-adjusted, ex ante, optimal technical trading rules in equity markets. *International Review of Economics & Finance* 12(1), 69-87.

[104] **Niederhoffer, V.** and Osborne, M. F. M. (1966). Market making and reversal on the stock exchange. *Journal of the American Statistical Association* 61, 897-916.

[105] **Nordin, P.**, Francone, F. and Banzhaf, W. (1995). Explicitly defined introns and destructive crossover in genetic programming. In: Rosca, J. P. (ed.), Proceedings of the Workshop on Genetic Programming: From Theory to Real-World Applications, 6-22, Tahoe City, CA.

[106] **Nordin, P.** and Banzhaf, W. (1995). Complexity compression and evolution. In: Eshelman, L. (ed.), Genetic Algorithms: Proceedings of the Sixth International Conference (ICGA95), 310-317, Pittsburgh, PA, Morgan Kaufmann.

[107] **Olson, D.** and Mossman, C. (2001). Cross-Correlations and Predictability of Stock Returns. *Journal of Forecasting* 20(2), 145-60.

[108] **O'Neill, M.**, Brabazon, A. and Ryan, C. (2002). Forecasting Market Indices using Evolutionary Automatic Programming. In: Chen, S.-H. (ed.), Genetic Algorithms and Genetic Programming in Computational Finance, Kluwer Academic Publishers.

[109] **O'Reilly, U.-M.** (1995). An Analysis of Genetic Programming. PhD thesis, Carleton University, Ottawa Carleton Institute for Computer Science, Ottawa, Ontario, Canada.

[110] **O'Reilly, U.-M.** and Oppacher, F. (1995). The troubling aspects of a building block hypothesis for genetic programming. In: Whitley, L.D. and Vose, M.D. (eds.), Foundation of Genetic Algorithms 3, Estes Park, Colorado, USA, Morgan Kaufmann.

[111] **Pardo, R.** (1992). Design, Testing, and Optimization of Trading Systems, John Wiley & Sons.

[112] **Pereira, R.** (2002). Forecasting ability but no profitability: An empirical evaluation of genetic algorithm-optimised technical trading rules. In: S.-H. Chen (ed.), Evolutionary Computation in Economics and Finance, 287-310, Physica-Verlag.

[113] **Pesaran, M.** and Timmermann, A. (1995). Predictability of stock returns: Robustness and economic significance. *Journal of Finance* 50, 1201-1228.

[114] **Potvin, J.-Y.**, Soriano, P. and Vallée, M. (2004). Generating trading rules on the stock markets with genetic programming. *Computers & Operations Research* 31(7), 1033-1047.

[115] **Price, G. R.** (1970). Selection and Covariance. *Nature* 227, 520-521.

[116] **Raj, M.** and Thurston, D. (1996). Effectiveness of Simple Technical Trading Rules in the Hong Kong Futures Market. *Applied Economic Letters* 3, 33-36.

[117] **Ratner, M.** and Leal, R.P.C. (1999). Tests of technical trading strategies in the emerging markets of Latin America and Asia. *Journal of Banking and Finance* 23, 1887-1905.

[118] **Rechenberg, I.** (1973). Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution, Fromman-Hozlboog-Verlag.

[119] **Reeves, C. R.** and Rowe, J. E. (2003). Genetic Algorithms - Principles and Perspectives: A Guide to GA Theory, Kluwer Academic Publishers.

[120] **Rosca, J. P.** (1997). Analysis of complexity drift in genetic programming. In: Koza, J. R., Kalyanmoy, D., Dorigo, M., Fogel, D. B., Garzon, M. H., Iba, H. and Riolo, R. L. (eds.), Genetic Programming 1997: Proceedings of the Second Annual Conference, Stanford University, CA, USA, Morgan Kaufmann.

[121] **Samuelson, P. A.** (1965). Proof that properly anticipated prices fluctuate randomly. *Industrial Management Review* 6, 41-49.

[122] **Scheinkman, J.** and LeBaron, B. (1989). Nonlinear Dynamics and Stock Returns. *Journal of Business* 62, 311-337.

[123] **Scholes, M.** (1972). The market for securities: Substitution versus price pressure and the effects of information on share prices. *Journal of Business* 45, 179-211.

[124] **Seyhun, H. N.** (1986). Insiders' profits, cost of trading and market efficiency. *Journal of Financial Economics* 16, 189-212.

[125] **Setzkorn, C.**, Dipietro, L. and Purshouse, R. (2003). Evolving Rule-Based Trading Systems. Technical report, ULCS-02-005, Department of Computer Science, University of Liverpool.

[126] **Sharpe, W. F.** (1966). Mutual Fund Performance. *Journal of Business* 39, 119-138.

[127] **Shiller, Robert J.** (1979). The Volatility of Long-Term Interest Rates and Expectations Models of the Term Structure. *The Journal of Political Economy* 87(6), 1190-1219.

[128] **Silva, S.** and Almeida, J. (2003). Dynamic maximum tree depth - a simple technique for avoiding bloat in tree-based GP. In: Cantú-Paz, E. et al. (eds.), Genetic and Evolutionary Computation - GECCO 2003: Genetic and Evolutionary Computation Conference, Chicago, IL, Proceedings: Pt. 2, Springer-Verlag.

[129] **Silva, S.** (2007). GPLAB - A Genetic Programming Toolbox for MATLAB. ECOS - Evolutionary and Complex Systems Group, University of Coimbra, Portugal. Manual + Software downloadable at http://gplab.sourceforge.net/download.html.

[130] **Sortino, F. A.** and van der Meer, R. (1991). Downside Risk. *The Journal of Portfolio Management* 17(4), 27-31.

[131] **Sortino, F. A.** and Price, L. (1994). Performance Measurement in a Downside Risk Framework. *Journal of Investing* 3(3), 59-64.

[132] **Sun, D.**, Benekohal, R. F. and Waller, S. T. (2003). Multi-objective Traffic Signal Timing Optimization Using Non-dominated Sorting Genetic Algorithm II. In: Cantú-Paz, E. et al. (eds.), Genetic and Evolutionary Computation - GECCO 2003: Genetic and Evolutionary Computation Conference, Chicago, IL, Proceedings: Pt. 2, Springer-Verlag.

[133] **Sweeney, J.** (1986). Beating the Foreign Exchange Market. *Journal of Finance* 41, 163-182.

[134] **Thomas, J. D.** and Sycara, K. (2002). Genetic Programming and the Predictive Power of Internet Message Traffic. In: Chen, S.-H. (ed.), Genetic Algorithms and Genetic Programming in Computational Finance, Kluwer Academic Publishers.

[135] **Tölke, F. W.** (1992). Exchange Rate Analysis with Point & Figure Charts, Peter Lang Verlag.

[136] **Tong, H.** (1990). Non-Linear Time Series: A Dynamical Systems Approach, Oxford University Press.

[137] **Tsang, R.** and Lajbcygier, P. (2002). Optimizing Technical Trading Strategies with Split Search Genetic Algorithms. In: Shen, S.-H. (ed.), Evolutionary Computation in Economics and Finance, Physica-Verlag.

[138] **Tsao, C.-Y.** and Chen, S.-H. (2004). Statistical analysis of genetic algorithms in discovering technical trading strategies. *Advances in Econometrics* 17, 1-43.

[139] **Tsay, R.** (1986). Nonlinearity Tests for Time Series. *Biometrika* 73, 461-466.

[140] **Vaihekoski, M.** (2009). A Note on the Calculation of the risk-free Rate for Tests of Asset Pricing Models. Working Paper. Downloadable at http://www2.lut.fi/~vaihekos/mv_research.html#wps.

[141] **Wang, G.** and Soule, T. (2004). How to Choose Appropriate Function Sets for Genetic Programming. In: Keijzer, M., O'Reilly, U. M., Lucas, S. M., Costa, E. and Soule, T. (eds.), Genetic Programming 7th European Conference EuroGP 2004, Proceedings, Volume 3003 of LNCS, 198-207, Springer-Verlag.

[142] **Wang, J.** (2000). Trading and hedging in S&P 500 spot and futures markets using genetic programming. *Journal of Futures Markets* 20, 911-942.

[143] **Whigham, P. A.** (1995). A schema theorem for context-free grammars. In: 1995 IEEE Conference on Evolutionary Computation, Volume 1, pages 178-181, Perth, Australia, IEEE Press.

[144] **Whigham, P. A.** (1996a). Search bias, language bias, and genetic programming. In: Koza, J. R., Goldberg, D. B., Fogel, D. B. and Riolo, R. L. (eds.), Genetic Programming 1996: Proceedings of the First Annual Conference, 230-237, Stanford University, CA, USA, MIT Press.

[145] **Whigham, P. A.** (1996b). Grammatical Bias for Evolutionary Learning, PhD thesis, School of Computer Science, University College, University of New South Wales, Australian Defence Force Academy, Canberra, Australia.

[146] **Yang, J. M.** (2003). An Evolutionary Approach for Molecular Docking. In: Cantú-Paz, E. et al. (eds.), Genetic and Evolutionary Computation - GECCO 2003: Genetic and Evolutionary Computation Conference, Chicago, IL, Proceedings: Pt. 2, Springer-Verlag.

[147] **Yu, T.**, Chen, S.-H. and Kuo, T.-W. (2004). Discovering Financial Technical Trading Rules Using Genetic Programming With Lambda Abstrac-

tion. In: O'Reilly, U.-M. et al. (eds.), Genetic Programming Theory and Practice II, Springer-Verlag.

[148] **Zadeh, L. A.** (1965). Fuzzy Sets. *Information and Control* 8, 338-353.