ON INCREASING THE SCOPE OF GENETIC PROGRAMMING TRADING AGENTS

by

Alexander Loginov

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

at

Dalhousie University
Halifax, Nova Scotia
June 2020

Table of Contents

List of Tables

List of Figures

Abstract

This research investigates the potential for widening the scope of Genetic Programming (GP) trading agents beyond constructing decision trees for buy-hold-sell decisions. First, both technical indicators (temporal feature construction) and decision trees (action selection) are co-evolved under the machine learning paradigm of GP with the benefit of setting Stop-Loss and Take-Profit orders using retracement levels demonstrated. GP trading agents are then used to design trading portfolios under a frequent intraday trading scenario. Such a scenario implies that transaction costs have a more significant impact on profitability and investment decisions can be revised frequently. Furthermore, existing long term portfolio selection algorithms cannot guarantee optimal asset selection for intraday trading, thus motivating a different approach to asset selection. The proposed algorithm identifies a subset of assets to trade in the next day and generates buy-hold-sell decisions for each selected asset in real-time. A benchmarking comparison of ranking heuristics is conducted with the popular Kelly Criterion, and a strong preference for the proposed Moving Sharpe ratio demonstrated. Moreover, the evolved portfolios perform significantly better than any of the comparator methods (buy-and-hold strategy, investment in the full set of 86 stocks, portfolios built from random stock selection and Kelly Criterion). Transaction costs (explicit and implicit or hidden) are important, yet often overlooked, attributes of any trading system. The impact of hidden costs (bid-ask spread) is investigated. The nature of bid-ask spreads (fixed or floating) is demonstrated to be important for the effectiveness of the automated trading system and a floating spread is shown to have a more significant impact than a fixed spread. Finally, the proposed GP framework was assessed on non-financial streaming data. This is significant because it provides the basis for comparing the proposed GP framework to alternative machine learning methods specifically designed to operate under a prequential model of evaluation. The GP framework is shown to provide classification performance competitive with currently established methods for streaming classification, and thus its general effectiveness.

Glossary

Hereafter the following terminology will be assumed [69].

Alpha	Alpha (the Greek letter α) is a term used in investing to describe a strategy's ability to beat the market, or it's edge. Alpha is thus also often referred to as excess return or abnormal rate of return, which refers to the idea that markets are efficient, and so there is no way to systematically earn returns that exceed the broad market as a whole. Alpha is often used in conjunction with beta (the Greek letter β).
Ask	Price at which broker/dealer is willing to sell. Same as Offer.
Balance	The value of your account not including unrealized gains or losses on open positions.
Bear Market	An extended period of general price decline in an individual security, an asset, or a market
Beta	A beta (the Greek letter β) coefficient is a measure of the volatility, or systematic risk, of an individual stock in comparison to the unsystematic risk of the entire market. In statistical terms, beta represents the slope of the line through a regression of data points from an individual stock's returns against those of the market.
Bid	Price at which broker/dealer is willing to buy..
Bull Market	A market which is on a consistent upward trend
Drawdown	The magnitude of a decline in account value, either in percentage or dollar terms, as measured from peak to subsequent trough
Equities	Ownership interest in a corporation in the form of common stock or preferred stock

Equity Market	An equity market is a market in which shares are issued and traded, either through exchanges or over-the-counter markets. Also known as the stock market, it is one of the most vital areas of a market economy because it gives companies access to capital and investors a slice of ownership in a company with the potential to realize gains based on its future performance.
Exchange	An exchange is a marketplace where securities, commodities, derivatives and other financial instruments are traded. The core function of an exchange is to ensure fair and orderly trading and the efficient dissemination of price information for any securities trading on that exchange.
Fibonacci Retracement	A Fibonacci Retracement is a popular tool among technical traders. It is based on the key numbers identified by mathematician Leonardo Fibonacci in the 13th century. In technical analysis, a Fibonacci retracement is created by taking two extreme points (usually a major peak and trough) on a stock chart and dividing the vertical distance by the key Fibonacci ratios of 23.6%, 38.2%, 50%, 61.8%, and 100%. Once these levels are identified, horizontal lines are drawn and used to identify possible support and resistance levels.
Fundamental Analysis	Macro or strategic assessment of where a currency should be trading on any criteria but the price action itself. The criteria often include the economic condition of the country that the currency represents, monetary policy, and other fundamental elements.
Futures	An obligation to exchange a good or instrument at a set price on a future date
Hit Rate	The hit rate is the ratio of the total number of winning trades to the total number of trades. It does not take into account how much was won, but simply if they were winners.

Kelly Criterion	The Kelly criterion is a mathematical formula relating to the long-term growth of capital developed by John L. Kelly, Jr. The formula was developed by Kelly while working at AT&T's Bell Laboratories. The formula is currently used by gamblers and investors for risk and money management purposes, to determine what percentage of their bankroll/capital should be used in each bet/trade to maximize long-term growth.
Leverage	The amount, expressed as a multiple, by which the notional amount of trade exceeds the margin required to trade.
Limit Order	An order placed with a brokerage to buy or sell a set number of shares at a specified price or better. Limit orders also allow an investor to limit the length of time an order can be outstanding before being canceled.
Market Noise	Price and volume fluctuations in the market that can confuse one's interpretation of market direction. Used in the context of equities, it is stock market activity caused by program trading, dividend payments or other phenomena that is not reflective of overall market sentiment. In general, the shorter the time frame, the more difficult it is to separate the meaningful market movements from the noise.
Money Management	The process of budgeting, saving, investing, spending or otherwise in overseeing the cash usage of an individual or group. The predominant use of the phrase in financial markets is that of an investment professional making investment decisions for large pools of funds, such as mutual funds or pension plans. Also referred to as 'investment management' and/or 'portfolio management'

Moving Average	A moving average (MA) is a widely used indicator in technical analysis that helps smooth out price action by filtering out the noise from random short-term price fluctuations.
Pattern	In technical analysis, the distinctive formation created by the movement of security prices on a chart. It is identified by a line connecting common price points (closing prices, highs, lows) over a period of time. Chartists try to identify patterns to try to anticipate the future price direction.
Pip	The smallest price increment in a currency. Often referred to as ticks in the future markets. For example, in EURUSD, a move of 0.0001 is one pip.
Pivot Point	A pivot point is a technical analysis indicator, or calculations, used to determine the overall trend of the market over different time frames. The pivot point itself is simply the average of the high, low and closing prices from the previous trading day.
Portfolio	A portfolio is a grouping of financial assets such as stocks, bonds, commodities, currencies and cash equivalents, as well as their fund counterparts, including mutual, exchange-traded and closed funds.
Profit	Profit describes the financial benefit realized when revenue generated from a business activity exceeds the expenses, costs, and taxes involved in sustaining the activity in question.
Return	A return, also known as a financial return, in its simplest terms, is the money made or lost on an investment over some period of time.

Sharpe Ratio	The Sharpe Ratio is a measure for calculating risk-adjusted return, and this ratio has become the industry standard for such calculations. It was developed by Nobel laureate William F. Sharpe. The Sharpe ratio is the average return earned in excess of the risk-free rate per unit of volatility or total risk.
Spot	Buying and selling forex with the current date's price for valuation, but where settlement usually takes place in two days
Spread	The distance, usually in pips, between the Bid and Ask prices.
Stock Market	The stock market refers to the collection of markets and exchanges where regular activities of buying, selling, and issuance of shares of publicly-held companies take place. Such financial activities are conducted through institutionalized formal exchanges or over-the-counter (OTC) marketplaces which operate under a defined set of government regulations. There can be multiple stock trading venues in a country or a region which allow transactions in stocks and other forms of securities. Both terms - stock market and stock exchange - are used interchangeably, the latter term is generally a subset of the former.
Stop-Loss	Also called SL. An order to buy or sell when the market moves to a specific price. A stop-loss order is designed to limit a loss when the price is moving in the opposite to the desired direction.
Support and Resistance	These terms are used by traders to refer to price levels on charts that tend to act as barriers, preventing the price of an asset from getting pushed in a certain direction.
Take-Profit	Also called TP. An order to buy or sell when the market reaches a target price to fix the profit.
Technical Analysis	Analysis applied to the price behaviour of the market to develop trading decision, irrespective of fundamental factors

Technical Indicator or TI	Any class of metrics whose value is derived from generic price activity in a stock or asset. Technical indicators look to predict the future price levels, or simply the general price direction, of a security by looking at past patterns. Examples of common technical indicators include Moving Average, Relative Strength Index, Stochastics, MACD, Bollinger Bands, etc.
Tick	A tick is a measure of the minimum upward or downward movement in the price of a security. A tick can also refer to the change in the price of a security from one trade to the next trade.
Trading Curb	A temporary restriction on program trading in a particular security or market, usually to reduce dramatic price movements. Also known as a collar or circuit breaker
Trading Strategy	A trading strategy is a method of buying and selling in markets that is based on predefined rules used to make trading decisions. A trading strategy can be likened to a trading plan that takes into account various factors for an investor.
Transaction Costs	Transaction costs are expenses incurred when buying or selling a good or service. In a financial sense, transaction costs include brokers' commissions and spreads, which are the differences between the price the dealer paid for a security and the price the buyer pays.
Win/Loss Ratio	The win/loss ratio is the ratio of the total number of winning trades to the number of losing trades. It does not take into account how much was won or lost, but simply if they were winners or losers.

Chapter 1

Introduction

1.1 The history of Exchanges and Equity Markets

The history of exchanges, marketplaces where various financial instruments are traded, extends back to antiquity. Available evidence suggest that even at 2000 years BC ancient traders in Babylon used notes and checks and that a currency exchange existed in Rome as early as second century AD [80]. The first known building that was specifically designed to be used as an exchange market was built in Barcelona in 1393 [80]. In 1608 the world's first important equity[1] or stock market[2] — the Amsterdam Exchange was founded. That was the biggest exchange by that time. The Amsterdam Exchange continued to grow and more than 4500 traders worked there every day at the end of 1722, thereafter the center of trading activity gradually moved to the London and Paris exchanges [80]. The first North American Exchanges, The New York Stock Exchange and Philadelphia Stock Exchange (the first US stock exchange), originated at the end of the 18th century [137].

The growth of Exchanges forced traders to develop ways to analyze markets and build trading rules — or so-called trading strategies[3] to maximize chances to gain a profit.

There are two major ways to analyze financial markets [92] — fundamental and technical analysis:

- Fundamental analysis. Reuters [137] defined fundamental analysis as follows: "Security analysis that seeks to detect misvalued securities through an analysis of the

---

[1] An equity market, also known as the stock market, is a market in which shares are issued and traded, either through exchanges or over-the-counter markets.

[2] The stock market or stock exchange refers to the collection of markets and exchanges where regular activities of buying, selling, and issuance of shares of publicly-held companies take place. These activities are conducted under a defined set of government regulations.

[3] A trading strategy is a method of buying and selling in markets that is based on predefined rules used to make trading decisions. A trading strategy can be likened to a trading plan that takes into account various factors for an investor.

firm's business prospects. Research [supporting decision making for Fundamental] analysis often focuses on earnings, dividend prospects, expectations for future interest rates, and risk evaluation of the firm. Information such as balance sheets, income statement, products, management and other market items are used to forecast a company's imminent success or failure, and hence the future price action of the stock." Thus, fundamental analysis is based on an analysis of the underlying economic conditions associated with a particular stock or currency and is not a part of this research.

- Technical analysis. Technical analysis uses historical prices to predict future price movements [92], and this thesis will focus entirely on technical analysis.

Technical analysis might be considered to be a 'timeless' method of market and price behavior such that a trader can make predictions about future market/price direction. However, there is no written evidence that technical analysis was used in ancient times [80]. The first known documented case of using technical analysis is a written set of trading rules, the "Sakata constitution". It was recorded in 18th century Japan by Sokyo Honma — a rice trader from Osaka.

One cornerstone of technical analysis is the candlestick chart [145, 139, 124, 80]. It was introduced at the same time as the "Sakata constitution." It is hard to believe that today's candlestick chart, as widely used by traders all around the world (and possibly now the most widely used charting technique), were only introduced to the Western world by Steve Nison who published a short article describing them in the Features Magazine in December 1989 [123] and then authored a book about candlestick charts in 1991 [124]. The candlestick summarizes the variance in price 'tick'[4] information over a selected period of time (typical values ranging from one second to one year) and includes the following attributes (Figure 1.1):

- Open and Close prices. The price at the beginning and at the end of the time interval respectively. The difference between Open and Close prices shows the direction of price movement during the time interval. If the Close price is higher than the Open price, then the body of the candlestick is normally colored in a light color and in a dark color otherwise.

---

[4] A tick is a measure of the minimum upward or downward movement in the price of a security. A tick can also refer to the change in the price of a security from one trade to the next trade.

- High and Low prices show respectively the highest and lowest extent of price movement within the selected time interval.

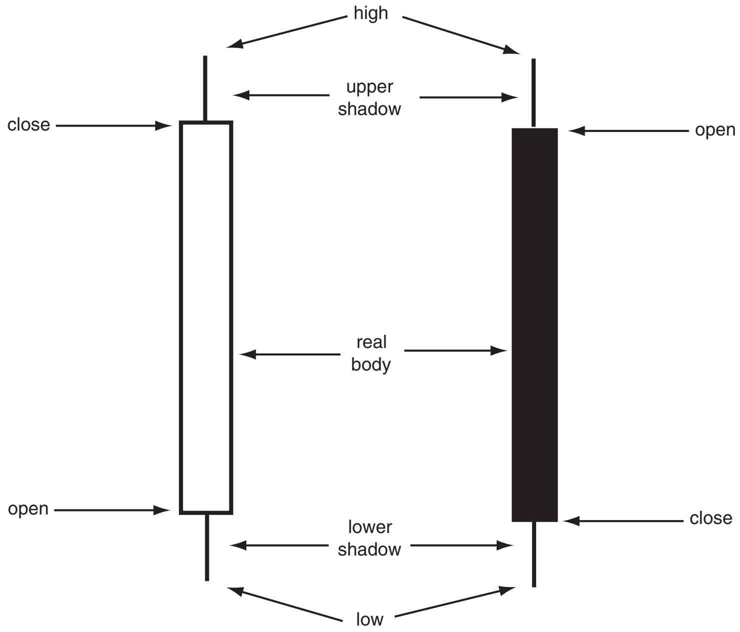


Figure 1.1: Candlestick attributes, adopted from [145].

Even though technical analysis history dates centuries back, what we know as a "modern" technical analysis was re-introduced at the end of the 19th century by Charles Dow. On July 3, 1884, Dow published the first stock index in the "Customer's Afternoon Newsletter." He calculated the price-weighted average by summing the prices of 11 stocks in the index and dividing by the number of stocks. That first index included nine railroads and two industrial stocks [80]. In the first half of the 20th century, the first technical indicators were created. As technical indicator (TI hereafter) is defined as a class of metrics whose value is derived from generic price activity in a stock or asset [69]. Technical indicators look to predict the future price levels, or simply the general price direction, of a security by looking at past patterns. In 1944 Leonard Ayers proposed a measure of business confidence — the advance-decline line (A/D line). Richard W. Schabacker, was the financial editor of Forbes magazine and of the New York Times when he began to recognize candlestick patterns. He was attributed as the first to use the terms "triangle," "pennant," (Figure 1.2) and “head-and-shoulders” (Figure 1.3) to describe what we are now considered well known chart formations [80].

Figure 1.2: Pennant Pattern. Pennant and Triangle chart patterns only differ by the duration and the appearance of a ‘flagpole’. Adopted from [69]

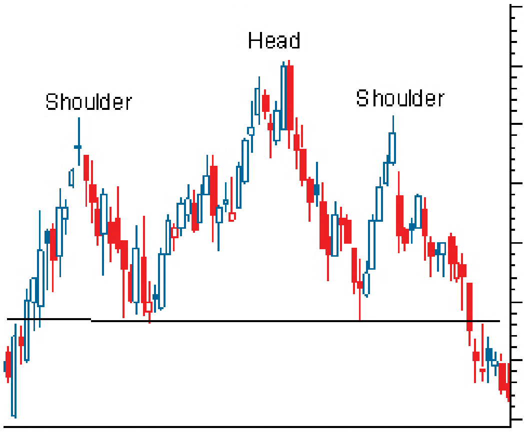


Figure 1.3: Head And Shoulders Pattern. Adopted from [69]

In 1948 Robert Edwards and John Magee published the first edition of Technical Analysis of Stock Trends. They demonstrated the technical patterns observed in hundreds of stocks. That book is still popular today and known as the “bible of technical analysis” (the ninth edition [50] was published in 2007).

In the 1970s, computers became widely available and traders started to use them to draw charts more quickly and accurately. In short, computers have changed the impact of technical analysis by minimizing the effort necessary to build them on a continuous basis. Conversely, some academics argued that technical analysis was impossible because prices were randomly distributed. This implies that price data had no history embedded in them that would be useful for predicting future prices. However, the popularity of technical analysis among traders has increased as access to powerful computers and higher-quality data improved [80].

1.2 Modern Technical Analysis

It is generally accepted that the modern technical analysis era started at the end of 19th century by the series of publications made by Charles Dow (Section 1.1). A fundamental assumption behind technical analysis is that price movement reflects all relevant information, and since human (trader) behavior tends to be repetitive, the price behavior can be repetitive as well [151]. Moreover, studying the historical price movements enables us to predict the future direction of price movement with a certain degree of confidence. More detailed basic assumptions behind technical analysis were presented by Robert Edwards et al. in their classic book "Technical Analysis of Stock Trends" [50] and summarized as follows:

- Prices are determined solely by the interaction of demand and supply.
- Prices tend to move in trends.
- Shifts in demand and supply cause reversals in trends.
- Shifts in demand and supply can be detected in charts.
- Chart patterns tend to repeat themselves.

In other words, traders who use technical analysis study the actions of the market itself, assuming that the market is always "right", often ignoring all other factors that can influence the market (e.g. political, social, natural disasters, etc.).

Technical traders utilize this approach on all markets and, generally speaking, we can divide the technical analysis into two major parts: the recognition of price patterns and price analysis with technical indicators. This thesis research mostly concentrates on the second part — the price analysis with technical indicators but, at the same time, assumes price "pattern recognition" using candlesticks as the starting point. In particular, the thesis investigates the ability of the genetic programming machine learning approach to automatically build profitable trading strategies for stock and currencies markets. Specifically, this thesis attempts to identify trading conditions that might undermine as well as improve the automatic operation of such an approach.

Price charts and technical indicators are available in most trading platforms and, the most commonly used TIs — such as Fibonacci Retracements, Stochastic Oscillator, Moving Average Convergence/Divergence (MACD), Moving Averages (MA), Relative Strength Index (RSI), and support/resistance levels — have proven valid in many instances [92]. Traders use TIs to draw out trends that enable them to make predictions regarding the future behavior of the market. Such predictions provide the basis for trading actions, e.g., buy, sell, or hold. *The combination of TIs with a decision rule provides the basis for a "trading strategy".*

There are hundreds of technical indicators nowadays, e.g. Colby in his "The Encyclopedia Of Technical Market Indicators" [31] provides a detailed description of almost three hundred different technical indicators. Furthermore, the number of existing technical indicators is constantly growing. Traders continue to create new technical indicators with the objective of maximizing profit and reducing losses. Some of these indicators become popular (e.g. Bill Williams' technical indicators [157, 60]) while others are only used by a few traders. If we also consider that each technical indicator has not one, but many possible ways of deploying it to make trading decisions (the trading rules), it becomes apparent that selection of a subset of technical indicators for designing a successful trading strategy is not a trivial task. In addition, it is worth noting that there are no formal rules that define how to select the most appropriate technical indicators, and every trader makes a choice based on experience [31].

Kirkpatrick and Dahlquist [80] noted that technical analysis is used in two major ways: *predictive or reactive*. The predictive approach assumes the goal of technical analysis is to predict future market state. Generally, experts in predictive analysis make money by selling their prognosis of the future market state to other traders. The predictive technical analysts include well-known experts in this domain (e.g. Bill Williams, the Profitunity Trading Group[5]). We know them because the publicity helps to sell their services.

On the contrary, cases in which technical analysis is deployed in a reactive mode (by commercial trading systems) are often concealed. Technical analysis, in the case of a reactive deployment, is used to react in real-time to the changing market conditions. For instance, such a deployment of a trading system can use a TI or a set of TIs with corresponding rules to define a moment when a position should be opened or closed. In other words, the trader is watching the market and *reacting* to when a specific technical condition is met. In that case, traders and investors make money

---

[5] `http://www.profitunity.com`

trading stocks or currencies for themselves or on behalf of their clients. In this case, institutions might be concerned that if many traders use the same strategy it could affect (change) the market behavior and reduce the effectiveness of the strategy. As a result, we know mostly about works from academia that describe various automated trading systems, but as will be shown below they are often based on assumptions that do not well represent the real trading conditions.

The popularity of technical analysis grew increasingly with the introduction of personal computers in the 1970s. On February 4, 1971, electronic trading was born. On that day, the first electronic stock market was founded — National Association of Securities Dealers Automated Quotation or NASDAQ. In 1987, the CME Group opened Globex trading system which gave access to electronic trading of treasury bonds, commodities, and currency exchanges [130]. The explosive development of information technologies at the beginning of the 21st century has boosted the significance of electronic trading even more. All leading stock exchanges made massive investments into information technologies and that accelerated the shift from the classic trading floors to electronic trading [130]. This shift to embrace information technology reflects an underlying trading advantage that trading using electronic means potentially provides. In April 2011, Robert Greifeld, NASDAQ CEO, was quoted as commenting: “It is over. The trading that existed down the centuries has died. We have an electronic market today. It is the present. It is the future” [19].

The introduction of electronic trading paved the road for the introduction of algorithmic or automated trading systems. Before giving an overview of automated systems and their classification, we need to look into the more general classification. Aldridge [2] has defined this as follows:

- Electronic trading — trading based on electronic transmission of trading orders.
- Algorithmic/automated trading — the systematic execution process — that is, the optimization of buy-and-sell decisions once these buy-and-sell decisions were formulated by another part of the systematic trading process or by a human portfolio manager.
- Systematic trading — computerized trading in which computer systems process real-time data and then formulate and go on to execute buy-and-sell decisions.

At the same time, Aldridge noted [2]: “To this day, the term ‘algorithmic trading’ usually refers to the systematic trading.” Based on that, this work uses the terms ‘algorithmic/automated trading’ and ‘automated trading systems’ interchangeably. That is to say, computerized trading is one in which computer systems process real-time data and then formulate and go on to execute buy-and-sell decisions.

Algorithmic/automated trading systems appeared in the 1970s, shortly after the introduction of electronic trading, and immediately became popular. But ever since, they have been surrounded by controversy about their effectiveness and influence on markets. For example, in 1986 Pamela [125] showed that some forms of trading programs (generally involving arbitrage between the stock market and the stock-index futures market) can contribute to sudden, sharp movement in stock prices. Tseng et al. [149] mentioned that “it has also been documented that algorithmic traders use their technological advantage to extract rent from other market participants and thereby increase their transaction costs.” Some sources [69] include factors such as the risk of mechanical failure and over-optimization.

Conversely, the number of positive arguments in support of automated trading significantly outnumbers the negative ones. Gomber [57] stated that the prevailing negative opinion about algorithmic trading, especially HFT, is driven in part by media reports that are not always well informed and impartial. Todorovic et al. [146] concluded that automated trading may reduce the impact of human emotions on decision making and can overcome problems that arise due to neglect or lack of concentration. Moriyasu et al. [119] found that algorithmic trading increases stock liquidity by narrowing spreads and increasing market depth. Furthermore, algorithmic trading increases commonality in liquidity at both high and low frequencies. Aggarwal et al. [1] also found that securities with higher algorithmic trading have lower liquidity costs, order imbalance, and order volatility, as well as evidence that higher algorithmic trading leads to lower intraday liquidity risk and a lower incidence of extreme intraday price movements. Vuorenmaa [154] discussed the pros and cons of automated high-frequency trading (HFT). By HFT, he implied automated trading executed at intra-daily intervals, but such that it excludes trading to minimize transaction costs. He mentioned that there is much confusion about HFT, regarding whether is it “good, bad, or ugly” In his work, the “Ugly” category mostly contained negative writings against HFT as reported by news media. In the "Bad" category, he included more detailed research arguments against HFT. Finally, he concluded, that surprisingly to non-professionals, the "Good" arguments outweigh the others. He stated that "Good" arguments are often ignored in media and should be brought to the fore of the discussion to be fair. Specifically, one of the most well known and most likely the "ugliest" argument against HFT is the Flash Crash of May 6, 2010. US stock market indices, stock-index futures, options, and exchange-traded funds experienced a sudden and unprecedented increase in volatility. However, Vuorenmaa noted that the conclusion the Securities and Exchange Commission and Commodity Futures Trading Commission (CFTC/SEC) came to, after the investigation (2010) was that "HFT firms did not trigger the crash, and cannot be blamed for it, but their trading response to a sudden selling pressure by an institutional seller exacerbated volatility." Kirilenko et al. [79] studied this event using E-mini S&P 500 stock index future data and came to the same conclusion.

Despite the controversy and often seen critique in popular media, the popularity of algorithmic trading has been steadily increasing ever since it was introduced, and in 2012 its market share was 85% (Figure 1.4).

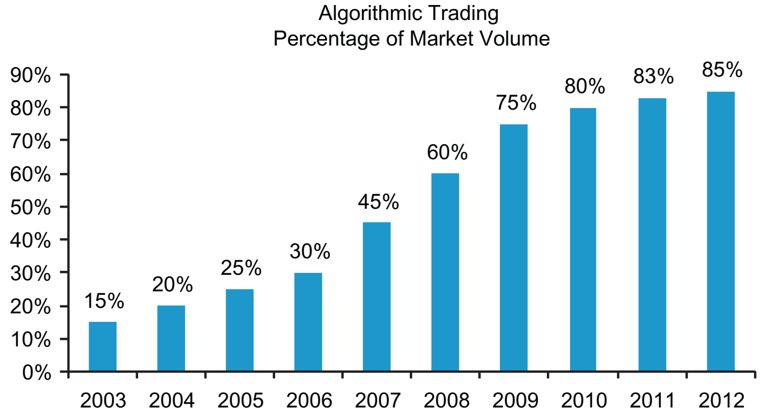


Figure 1.4: Algorithmic Trading Percentage of Market Volume. Adopted from [81].

Independent market analysts predict that the market size of algorithmic trading will continue to grow at a Compound Annual Growth Rate[6] (CAGR) of 11.1% (the

---

[6] Compound annual growth rate (CAGR) is the rate of return that would be required for an investment to grow from its beginning balance to its ending balance, assuming the profits were reinvested at the end of each year of the investment's lifespan. [69]

prediction horizon covers 2019 – 2024 years) [111].

The first automated trading algorithms simply replicated human traders' behavior and utilized existing trading methods and technical indicators: Various Moving Averages (simple, weighted and exponential), Relative Strength Index, Bollinger Bands, etc. Since then they have became significantly more autonomous (capable). Various technical analysis methods are commonly used for data mining and preprocessing the assets' prices, and new techniques have been developed to improve the performance of the automated trading systems. The two general approaches of PairS Trading (PST) and Machine Learning (ML) have received the most attention [67].

Pairs trading. PST is in everyday use by many market participants, such as hedge funds and other investors who rely on a 'Risk arbitrage' investment strategy [47]. Robert Kissell [81] described pairs trading as follows: "A simple statistical arbitrage strategy is pairs trading which aims to capitalize on the imbalances between two assets in the hope of making money once the imbalance is corrected. Pairs trading is a pure relative value strategy between two (possibly more) assets: Given two securities which historically moved together, take a long-short position as they diverge and realize a profit as the spread, the mispricing dynamics, converge back to the long-run mean." Even though PST is a long-term strategy, it can be effectively applied for short time windows and is an important part of High-Frequency Trading[7] (HFT) [5].

Machine Learning. Financial markets' price data can be described as a data stream that is fundamentally non-stationary, non-linear and noisy [67, 68, 148, 159]. Traditional statistical methods fall short in that case because they assume that the process driving data stream content is linear and (or) stationary. Machine learning algorithms have the potential to address this issue [67]. In this regard, Artificial Neural Networks (ANN) and Genetic Algorithms (GA) are among the most often seen machine learning methods used as the basis for automated trading systems.

Automated trading systems have made a huge step forward over the years and became a dominant technology in financial markets (Figure 1.4). Still, they have not solved all the problems and have not answered all the questions. There are always

---

[7] There is still a controversy about the definition of High-Frequency Trading (HFT). Different authors define it differently. In this case, the author defines it as follows: "HFT is the use of computers to trade very quickly and at high speed."

challenges that automated trading will have to overcome and solutions to be found. Some of the challenges are new and arise from automated trading itself, and some belong to technical analysis that lies in the base of automated trading. Challenges of particular relevance to the development of this thesis include:

1. Results of testing of many algorithmic/automated trading systems show that classical technical analysis of historical price feeds does not guarantee positive results. Park et al. [127] reviewed in their 2004 report 92 academic studies published between 1986 and 1990 years that tested the profitability of technical analysis strategies. Fifty-eight of reviewed studies concluded that positive results could be gained from using technical analysis. Still, twenty-four of the studies found that the use of technical analysis led to negative outcomes.

2. Supply and demand are among the fundamental forces that drive market prices, and the stock market is no exception. If many traders use automated systems that rely on the same set of traditional technical indicators and trading rules and that generate the same trading signals, it can affect the supply and demand and, therefore, change the market behavior and price movement.

3. There can be significant reduction to the timespan during which newly developed/evolved trading systems/rules will stay productive.

4. Technical analysis is based on the assumption that human behavior tends to be repetitive [50, 80, 139]. In contrast, the behavior of some machine learning techniques, such as evolutionary methods, might be unpredictable and not repetitive, such that makes the already challenging task of predicting the future price movement even more difficult.

This work relies on Genetic Algorithms and Genetic Programming[8] in an attempt to address the mentioned challenges and build an effective automated trading system for stock and foreign exchange markets. The specific approach adopted will be to coevolve both decision trees (DT) and technical indicators (TI) simultaneously. Thus, a unique TI-DT coupling is evolved relative to the current market situation. It also

---

[8] Under this application domain, the end result is a model, and therefore the distinction between GA and GP is not important.

links the fitness expressed at the level of DT to the TI without having to define any surrogate performance functions for the TI.

In brief, this thesis research attempts to address the mentioned challenges by adopting the following combination of mechanisms:

- Coevolution of decision trees and technical indicators allows unique technical indicators to evolve and, therefore, provide the basis for trading behavior to appear that is different from automated trading systems that rely on traditional TIs and trading rules. This helps to address challenges 1 and 2.

- The introduction of Stop-Loss[9] (SL) and Take-Profit[10] (TP) orders and their verification and adjustment with the support and resistance levels[11] helps to address challenge 1. In particular, if the market's volatility does not experience rapid and intense movements, the system operates in predictive mode — open positions are closed by the limit orders alone. Otherwise, it evolves trading agents to react to the price movement quickly — open positions are closed by the limit orders and (or) trading signals.

- Retraining criteria are introduced to address challenges 2 and 3. The retraining criteria are a set of three metrics that enables the continuous performance monitoring of the deployed trading agent. If its performance falls below the target level a retrain process is initiated and a new trading agent is evolved.

- The introduction of a multi-agent version of the FXGP algorithm is investigated for improving the overall quality of trading signals generated by the system (challenge 1).

- The stock selection algorithm significantly improves returns generated by the proposed automated trading system and is intended to address challenges 1–3. Also, its ability to quickly switch between trading assets helps to address challenge 4.

---

[9] Also called "SL". An order to buy or sell when the market moves to a specific price. A stop-loss order is designed to limit a loss when the price is moving in the opposite to the desired direction.

[10] Also called "TP". An order to buy or sell when the market reaches a target price to fix the profit.

[11] Support and resistance levels are used by traders to refer to price levels on charts that tend to act as barriers, preventing the price of an asset from getting pushed in a certain direction.

- Understanding the effect of hidden trading costs (Bid-Ask spread)[12] on the effectiveness of automated trading systems. This is a factor frequently ignored, but underlies challenges 1 and 3.

The thesis is structured as follows:

- Chapter 2. Provides an overview to related work that use different methods of indicator selection and their incorporation into different automated trading systems. In essence there are two basic approaches to constructing automatic trading agents:

    1. Accept a set of prior man-made TIs and deploy a machine learning algorithm to identify the conditions under which buy, hold, sell orders are placed.
    2. Attempt to identify parameters for man-made TI using an optimization algorithm and then deploy a machine learning algorithm to identify the conditions under which buy, hold, sell orders are placed.

  In this work GP is used to *simultaneously* construct TI and rules to define conditions under which buy, hold, sell orders are placed. The approach to achieve this is based on genetic programming and coevolve both TIs and trading rules (DTs) simultaneously in order to minimize the impact of prior decisions.

- Chapter 3 provides an overview of the core part of the proposed automated trading system - a genetic algorithm (ForeX Genetic Programming or FXGP hereafter) that evolves trading agents able to generate trading signals (buy, sell or hold). [97, 98, 99].

    - Section 3.1 provides an overview of the first version of the FXGP algorithm. Innovations specific to the approach pursued by this work may be highlighted as follows:

        1. Simultaneous coevolution of TI and trading rules. Very few researchers have managed to perform both tasks simultaneously. In doing so we

---

[12] The difference between the Bid and Ask prices, where this has an impact on the actual risk as experienced by a trading agent, autonomous or otherwise.

effectively avoid introducing unwanted biases into each step. Naturally, the trading rules are dependent on the quality of the TI, which in themselves represent a process of temporal feature construction.

2. Validation partition of data. The purpose of validation is to identify a single 'champion' trading agent for performing trading.
3. Retrain criteria. The following three quality criteria to monitor and identify the moment when the current trading agent should be replaced were introduced: drawdown[13], the number of consecutive losses and the number of consecutive candlesticks without trading activity.
4. Support for tuning of Stop-Loss orders. SL orders are used to define limits on how much loss is acceptable before pulling out of a stock.

This section also includes performance evaluation and critique of the first version of the FXGP algorithm.

- Section 3.3 addresses the issues (Section 3.2) that were discovered after the evaluation of the first version of the FXGP algorithm (Section 3.1). A following set of changes has been introduced to the first version:
    1. The original set of TI functions that contained seven instruction types was reduced to three instructions. Multiplication that produces a TI with the wrong scale, two division functions and a square root function that frequently resulted in illegal operations (e.g. division by zero or square root of negative value) were all dropped from the instruction set. That removed instructions that typically resulted in intron behavior.
    2. Originally three types of TI were supported: Value, MA and WMA, whereas the new set of the TI parameters contains only two types: Value and MA.
    3. The decision tree (DT) intron nodes detection and removal.
    4. The Take-Profit orders were introduced in addition to SL orders.
- Section 3.4. Evaluation of a prior TI with evolved trading rules under a prior portfolio versus coevolution of both TI and trading rules (FXGP)

---

[13] The magnitude of a decline in account value, either in percentage or dollar terms, as measured from peak to subsequent trough

under the same portfolio. Runs are repeated 100 times in each case and the consistency of the outcome when using coevolution is shown to be higher than without. Moreover, the region of profitability for each system is distinct.

- Chapter 4. Evaluation of the FXGP coevolutionary framework under a completely different application domain — predicting the movement in electricity utilization data. The objective of this is two fold: 1) permit simplification of FXGP as various features of potential utility to stock markets are no longer necessary; and, 2) enable comparison with streaming classification algorithms that are also appropriate to this simpler task. Within this context including the candlestick 'pre-processing' provides a significant benefit over that of current practice (which deploys the streaming algorithm directly to the original data). Significantly better accuracy than available from the Naive Bayes and Hoeffding Tree algorithms (available from the well known MOA toolkit) was also achieved. This is a particularly important result, because Hoeffding Trees also attempt to identify temporal features (from the stream) as well as constructing a decision tree for the purpose of describing classification rules, i.e. properties that are shared with those from FXGP. Finally, this work also draws attention to the capacity of FXGP to operate under real-time conditions. Construction of an entirely new solution (i.e., 1000 generations) is concluded within 2.5 seconds and typically less than 1.5 seconds. This work appeared at a Special Topics session at IEEE WCCI 2016 [101].

- Chapter 5. This chapter presents the results of the performance improvement of the proposed FXGP algorithm by introducing and assessing an approach for limit orders (SL and TP) verification with the support and resistance levels. The investigation of three types of support and resistance levels — Moving Average[14], Pivot Point[15] and Fibonacci Retracement[16], has shown that SL and TP orders verification based on Fibonacci Retracement significantly improves the performance and outperforms two other methods. This work appeared at a Special Topics session at IEEE 2015 [100].

  - Chapter 6 represents the capability to migrate funds within the portfolio[17] of stocks during trading. This work assumes that the stocks constituting such a portfolio have been defined a priori (e.g. stock available over the trading period, and traded at a sufficient frequency). The main interest is in demonstrating the capacity for moving funds around such a portfolio automatically in real-time under a intra day 'frequent trading' environment (minute-to-minute trading). Some of this work was accepted to appear in [102].
  - Chapter 7. This chapter investigates the influence of the Bid-Ask spread on the performance of the proposed automated trading system. The results show that the importance of Bid-Ask spreads is significantly underestimated and that floating spreads can dramatically degrade the performance of automated trading. Recommendations are made for identifying conditions under which the impact of Bid-Ask spreads can be reduced (lower frequency of trading in particular).

- Chapter 8. Summarizes the contributions of this thesis research.

---

[14] A moving average (MA) is a widely used indicator in technical analysis that helps smooth out price action by filtering out the "noise" from random short-term price fluctuations.

[15] A pivot point is a technical analysis indicator, or calculations, used to determine the overall trend of the market over different time frames. The pivot point itself is simply the average of the high, low and closing prices from the previous trading day.

[16] A Fibonacci Retracement is a popular tool among technical traders. It is based on the key numbers identified by mathematician Leonardo Fibonacci in the 13th century. In technical analysis, a Fibonacci retracement is created by taking two extreme points (usually a major peak and trough) on a stock chart and dividing the vertical distance by the key Fibonacci ratios of 23.6%, 38.2%, 50%, 61.8%, and 100%. Once these levels are identified, horizontal lines are drawn and used to identify possible support and resistance levels.

[17] A portfolio is a grouping of financial assets such as stocks, bonds, commodities, currencies and cash equivalents, as well as their fund counterparts, including mutual, exchange-traded and closed funds.

Chapter 2

Background and Related Work

The proposed automated trading system and its components are designed provide the basis for answering questions about the deployment of automated trading agents. As such the approach to designing a generic automated trading agent for operation under different markets and frequency of trading. Due to the number of covered tasks and different nature of the source data (financial and non-financial data streams), this chapter will be divided into separate sections. Each of them will provide background and overview works related to the respective Chapters 3–7. The approach adopted, by necessity, is therefore quite specific to GP. This does not mean that other approaches are not relevant, but that our objective to to identify a process for automating the design of training agents such that it is then possible to move forward and answer questions about the significance of transaction costs, asset selection heuristics, and the impact of bid-ask spreads on the automated trading agent.

2.1 Genetic Programming and Non-stationary Data Streams

GP constitutes an approach to model building in which multiple candidate solutions are maintained (the population) [84, 117]. Each candidate solution is expressed as a **genotype** and requires decoding into a **phenotype** (expressing a program) before it can be applied to a task, such as expressing a TI or DT. For example, in the case of a linear representation, the genotype is described in terms of a sequence of integers or genes [21]. Each integer value is decoded into a legal instruction and executed on a virtual machine (i.e. the ‘fetch-decode-execute’ cycle is simulated). Prior decisions are necessary in terms of defining what instructions to support. Indeed, this thesis will later show that a simpler instruction set can be adopted than was previously assumed (i.e. [98, 97]), resulting in faster evolution, without negatively impacting on the quality of the trading agents. **Selection operators** define which individuals become ‘parents’ therefore producing new candidate solutions that form the basis for a new population as well as defining which individuals ‘survive’ between consecutive training epochs or generations. The selection operator may use performance criteria to guide this process. **Variation operators** modify the genotype of parents to define new candidate solutions or ‘offspring’. Such operators either exchange sequences of the genotype between two parent individuals (crossover) or modify specific genes (mutation). **Crossover** therefore creates new programs through a process of mixing material that currently exist in the population, whereas **mutation** may introduce (genetic) material that does not currently exist in the population.

Unlike most machine learning algorithms, evolutionary methods need not enforce tight constraints between representation, performance function, and credit assignment. This means that evolutionary methods have the potential to adopt performance objectives specific to the task domain. Moreover, adopting a co-evolutionary approach provides the basis for mixing multiple representations, so increasing the range of tasks that can be solved simultaneously. For example, in constructing both TI (as opposed to merely selecting from a set of predefined TI) and DT, performance is only directly expressed at the level of the DT. A **symbiotic coevolutionary** approach was previously proposed to link the fitness expressed at the level of DT to the TI (Figure 2.1) without having to define surrogate performance functions for the TI [98]. This is important because such surrogate performance measures can introduce unwanted biases (limitations) to the type of DT discovered. In this thesis the same coevolutionary approach will be assumed as the starting point. However, this will be extended to support multi-agent operation, stop-loss and take profit orders, and more efficient execution. The latter is particularly important because without this, real-time operation at minute-to-minute intervals would not be possible.

In attempting to place GP within a wider context of what is necessary to efficiently construct models under environments displaying non-stationary properties, three generic properties are often identified [40, 63]: Evolvability (plasticity), memory, and diversity. **Evolvability (plasticity)** reflects the efficiency with which ‘useful’ phenotypic variation is generated from the current state of the environment.[1] One of the properties that contributes to the evolvability of a representation in GP is

---

[1] The ‘state’ of the environment is characterized by the sample of price information used to identify a trading agent. This typically takes the form of a finite sample of the most recent historical data, quantized in the form of a sequence of candlesticks (see Chapter 1).

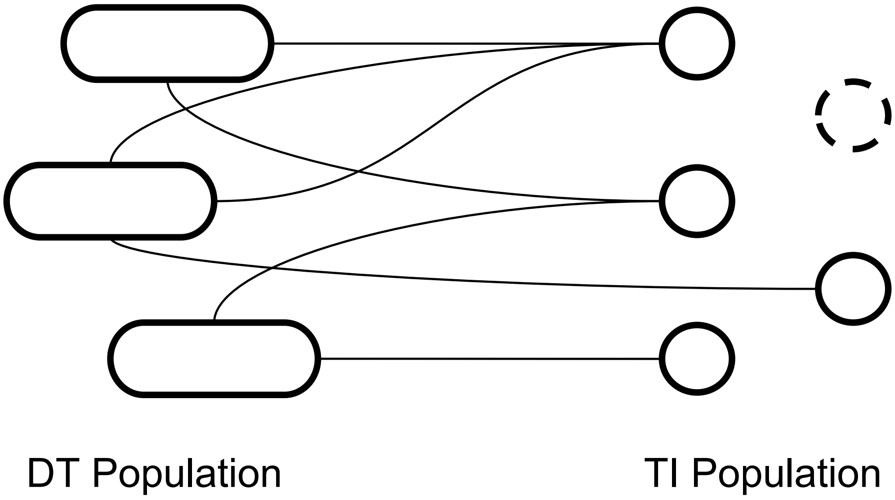


Figure 2.1: DT–TI interaction, symbiotic co-evolution. Fitness is only directly expressed at the DT population. Each DT may index a subset of TI. Each TI can appear in more than one DT. If a TI fails to be used by any DT it is considered not useful and deleted.

support for modularity, especially under non-stationary tasks [152, 63]. Under the trading agent task, the division of duties between identification of suitable TI and DT results in a natural metaphor for modularity. That is to say, the proposed system is not merely choosing between a predefined set of TI, but evolving a unique set of TI (relative to the current characterization of price information). Specifically, it simultaneously coevolves both the TI and DT in independent populations (Figure 2.1). Thus, as effective TI are discovered, their variants more frequently reproduce within the TI population. However, performance is only ever expressed through the quality of the DT, to do otherwise would require the introduction of a surrogate metric for the true performance objective. Hence, TI quality is only ever a function of their utilization by individuals from the DT population. Moreover, maintaining independent populations for TI and DT allows one to freely adopt representations appropriate to each task.

Memory refers to how multiple sources of genotypic information are recombined to construct new candidate solutions. This provides the basis for pursuing multiple solution properties simultaneously and is central to how evolutionary methods go about discovering solutions [136]. **Diversity** under non-stationary environments is considered through both the scheme adopted to interface FXGP to the data and the action of the variation operators. In the case of the FXGP interface, it adopts a change detection approach [97, 63]. That approach assumes that price data is described by a non-stationary process in which changes can be detected by (multiple) criteria describing poor trading outcomes relative to the trading agent currently deployed. On detecting poor trading behavior the FXGP training process is re-triggered from a completely new (set of) initial DT–TI population.

2.2 Discovering Trading Agents Using GP (Chapter 3)

The Foreign Exchange Market (Forex) is the world biggest financial market which produces 1/3 of all financial transactions in the world [120]. The average daily turnover of Forex was almost \$4 trillion in 2010 and it was 20% higher in April 2010 than in April 2007 [140]. The global Forex market daily turnover hit \$6.6 trillion at the beginning of 2020 [53]. A Forex market consists of currency pairs which are weighted by economic conditions for that specific denomination versus any or all others in the marketplace. Thus, the perceived value of a currency is a reflection of the market's ranking for that denomination's economy on any given day. An Forex market is technically 'purer' than a stock market, i.e. a currency price action reacts more strongly to resistance and support levels than equity market do [129]. All the above factors make Forex markets very attractive for traders and expand the demand of automated trading systems, albeit under demanding conditions.

Neely at al. [121] examined intraday trading strategies using four currency pairs — USDDEM, USDJPY, GBPUSD, and USDCHF using historical half-hour bid and ask quotes for spot Forex rates over 1996. They compare two methodologies, a genetic program that can search over an extensive class of (possibly nonlinear) trading rules and linear forecasting models. *They concluded that when assuming realistic transaction costs and trading hours, neither of the methods show evidence of excess return.*

Mendez at al. [115] used a genetic algorithm to optimize a set of trading rules that constituted a trading system for the Forex market. They used a set of classical technical indicators and trading rules, such as long and short term averages, position in a trading range, etc. They evaluated the proposed trading system over 2006–2010 using two currency pairs (EURUSD and GBPUSD) and four different time intervals

(1, 5, 15 and 60 minutes). *They concluded that the developed strategies showed difficulty in achieving positive results if transaction costs were taken into account.*

Godinho [56] used a genetic algorithm for parameter optimization of traditional trading strategies based on four popular technical indicators: Exponential Moving Average (EMA), Relative Strength Index (RSI), the Stochastic Oscillator (%k) and a three-period moving average of the stochastic oscillator (%d). The algorithm was tested on four currency pairs — USDHKD, USDSGD, EURUSD, and GBPUSD using 5 and 15 minute averages of tick-by-tick prices from September 2008 to February 2011. The results of the four mentioned currency pairs were compared with each other. He concluded that the proposed genetic algorithm was to be able to generate positive returns in the case of the USDSGD, even with transaction costs taken into account, *but could not reliably do so for other currency pairs.*

Cirillo et al. [30] proposed a Genetic Programming architecture to generate Forex trading strategies. They evolved free-form strategies that did not rely on any prior models. The system was tested on 5 minute candlesticks of four currency pairs (AUDUSD, EURUSD, GBPUSD and USDJPY). They considered only explicit transaction costs matching those applied by Citigroup on live trading: 15 USD were applied per every million USD transacted, appropriately converted for non-USD based currency pairs. They declared the overall best annual return at the level of 19%. However, the authors mentioned that "Performing a comparison between works in this field is not straightforward, given the many variable aspects involved in automated trading strategies and systems."

Manahov et al. in [110] offered Forex forecasts by applying a special adaptive form of the Strongly Typed Genetic Programming (STGP) to currency pairs. The STGP forecasting performance was compared with linear forecasting models. They found evidence of statistically significant excess return even with appropriate transaction costs after applying their system to 5 minutes data of six currency pairs: EURUSD, USDJPY, GBPUSD, AUDUSD, USDCHF and USDCAD.

Vasilakis et al. presented in [153] a genetic programming trading technique in an attempt to forecast the next day's return for the EURUSD currency pair. They compared their system with three traditional strategies (naive strategy, MACD, and a buy-and-hold strategy) and a hybrid evolutionary artificial neural network. They applied transaction costs of 1 pip per trade, and returned the most profitable trading strategy with the proposed method.

In short, trading costs can be a significant source of uncertainty in deploying automatic trading strategies for Forex markets, on account of the bid-ask spread (Chapter 1). Additionally, the underlying data describing such trading environments is typically non-stationary. Thus, assuming the classical approach of training over fixed partitions of data (e.g., training, validation and test) results in brittle solutions that could be specific to the partition on which they are evolved [40].

One solution proposed to this problem is to train on a continuous basis — a 'rolling window' approach [68, 115], while explicitly maintaining population diversity [40] in which a sequence of data (or window) is used for training (N_t) and the champion individual is deployed as the trading agent for the next δ data points. On reaching the end of the trading period the training window of N_t is realigned with the end of trading and training recommences using the content of the current population. Questions potentially arise regarding the number of generations to achieve best performance [40]. This is related to the degree of non-stationary (or appropriateness of any model bias) associated with the transition between training and trading periods. Indeed other authors have adopted training at *every* sliding window location [158]. Another approach might be to coevolve a subset of the training partition [114] or to combine coevolution of the training partition with an explicitly streaming context [7].

The approach adopted by this thesis assumes that retraining of the automated trading agent is explicitly triggered by performance of the champion agent. That is to say, when performance of a previously satisfactory model degrades below a threshold, the process generating the data has probably changed. This approach was previously benchmarked under a currency exchange setting [97, 98], and informed by developments under streaming data [63]. In adopting multiple performance objectives, it was possible to more closely reflect the multi-criteria nature of trading agent operation [97, 98].

A third theme that potentially impacts on the performance of autonomous trading strategies is the design of the technical indicators themselves. Specifically, the design of TI has a considerable impact on how the trading data is 'interfaced' to the trading data. Thus, not only the type of TI, but the parameterization of the TI need to be considered [52]. Some previous research has been reported with respect to the evolution of technical indicators (TI) alone [52]. One of the goals of evolving TI independently is to provide parameterizations for basic statistics such as moving averages, that are particularly appropriate for the data in question [155]. More recently separate TI were evolved for buying and selling [64]. Several authors have evolved trading rules or decision trees (DT) relative to a set of a priori selected TI [41, 66]. One of the most well known instances of GP in this context is 'EDDIE' [147]; the resulting IF–THEN rules are sampled by a human trader to determine which to use. Most recently, systems have been proposed to first evolve parameters for a fixed set of TI with the identification of DT e.g., [68]. To do so, a representation is assumed in which all of the information for the TI and DT appear in the same genome. This work considers this problematic as both components need to be correct for an individual to be successful. As emphasized by the study [40], it is recognized that under non-stationary environments a modular representation can be beneficial. There are many ways of potentially achieving this. This work adopts a symbiotic approach [46] to coevolve trading indicators together with trading decision trees. Thus, the crafting of uniquely appropriate rules and technical indicators for the current underlying market dynamic is supported. This led to the development of FXGP [97, 98, 99] which will represent the starting point for the framework ultimately proposed by this research (Chapter 3).

One final topic worth recognizing is that, as an explicitly temporal task, then reinforcement learning approaches have been considered for automating the design of trading agents [118]. Reinforcement learning explicitly designs a reward function such that an agent is rewarded for maximizing the cumulated reward over a possibly infinite time horizon [144]. However, this says nothing about the structure of the agent or its ability to represent temporal properties (the emphasis of the approach to designing agents taken in this thesis). Reinforcement learning is typically formulated as a process for continually updating the free parameters of a model through the method of temporal differences (a gradient method). Conversely, approaches based on evolutionary computation that attempt to address reinforcement learning problems typically assume an episodic formulation in which the model is only updated after some end condition is encountered [156].

In summary, adopting a genetic programming approach to constructing trading strategies provides the basis for answering questions about which inputs to use as well as how to construct a model. This property is shared by other machine learning approaches, such as decision tree induction. Interestingly, decision tree methods are rarely applied to the financial trading application domain, but have been widely applied to the more general topic of streaming data analysis. Moreover, the streaming data scenario also represents a setting in which the underlying process might be non-stationary.

2.3 Non-Financial Streaming Data Analysis (Chapter 4)

The goal of financial forecasting is to make predictions regarding the state of the market at the next time step, i.e. will the market move up, down or experience no (significant) change [74, 75]. From a wider perspective, this general requirement appears in prequential classification tasks under streaming data [15, 54]. Specifically, a machine learning model is attempting to predict the direction of a feature at the next time step before the direction is actually known. However, as the feature advances (i.e. the outcome is known), the model is allowed to perform an update (it does not necessarily have to update). A widely used application example of this is predicting whether the consumption of a utility (e.g. water, gas, electricity) will increase or decrease at the next time step relative to the recent past.

Despite the wide range of research conducted in evolving agents for financial trading and forecasting tasks, there has not been as much interest in applying such frameworks to the related task of streaming data classification [63]. Recent noteworthy examples include Vahdat et al. [152] who applied GP to streaming data under finite labelling budgets. They assumed the Symbiotic Bid-Based GP (or SBB) approach to coevolving programs into teams [91]. They showed that the GP can be successfully applied to streaming data classification tasks and highlighted two factors of particular significance: 1) active learning can be used to decouple GP from the immediate stream content, with the objective of manipulating the distribution of classes, 2) support for the coevolution of teams of programs implied that it was easier to react to the changing dynamics of stream content than classifiers defined by single programs.

Khanchi et al. [77] built on this SBB streaming framework for botnet detection under data label budgets and very imbalanced class distributions. The results were compared to Naive Bayes and Hoeffding tree classifiers from MOA framework [17] employing thirteen datasets from the CTU-13 network security dataset collection. Specifically, it was demonstrated that active learning heuristics could be designed to develop strong classification performance under low levels of label budgets. In both cases, however, each exemplar is entirely self contained (i.e. exemplar order is not important). In short, there is no requirement to construct temporal features capable of capturing relationships between sequences of inputs. This is not the case in the trading scenarios encountered in this work. Indeed, this represents a central requirement of this thesis. Indeed, several works that use GP in forecasting applications do not address the issue of temporal feature construction, but instead assume all the data to be simultaneously available and concentrate on the regression task alone, e.g. [36], [25].

Requiring GP to construct temporal features might be avoided if instead a preprocessing step is performed in which the original temporal data is described using temporal features. For example, Khanchi et al. used 'flow' based pre-processing of network data [77] and Maheswaran and Khosa assume Wavelets [107]. Indeed, even Deep Learning approaches to trading agent design currently make extensive use of feature preprocessing [12]. Conversely, a sliding window of the last 'n' exemplars might be assumed, with GP then learning how to index such exemplars to construct temporal features [142]. At the time of writing this thesis a 'generative' formulation of GP was proposed in which only the current exemplar $x(t)$ is input to the model, and the use of memory 'internal' to GP used to develop recurrent properties that were sufficient for predicting time series sequences [76]. In the case of financial data, and this thesis in particular, we adopt the case of previously calculated TI as an example of temporal feature construction through pre-processing, and perform an ablation study using GP with and without the proposed temporal feature construction approach.

In summary, both stock and Forex markets represent environments in which the underlying process is potentially non-stationary, implying that it is particularly important to address the issue of when to rebuild agents. In the context of streaming data classification, making decisions under non-stationary tasks is frequently equated with ‘shift’ and ‘drift’ [63, 44]. Models may either take the form of ensembles of multiple decision makers and (or) incrementally react to each and every sample from the stream. A body of research has also been developed for characterizing stream content statistically and relating this to change detection (e.g. Hoeffding Trees) [54, 15]. This provides a unique opportunity to compare the utility of the proposed approach for coevolving TI–DT with statistical methods for constructing decision trees under prequential streaming classification tasks. Thus, the utility (or otherwise) of the coevolutionary approach can be established outside the specifics of trading agent functionality which are often difficult to replicate, yet significant to financial applications.

2.4 SL and TP Orders Verification With Fibonacci Levels (Chapter 5)

The goal of this part of the thesis is to extend the proposed FXGP framework to incorporate the use of ‘retracement,’ or the tendency for a financial asset’s price to move between an upper and lower bound. The upper bound is known as the ‘resistance’ level, and the lower bound is known as ‘support.’ The identification of support and resistance levels characterize the direction of movement of a price and therefore form the basis for predictive sell or buy strategies. The support and resistance levels are used for the purpose of creating stop-loss (SL) and take profit-orders (TP). Specifically, support levels define a price at which downward trends do not pass beyond, and hence they appear as if they bounce off the support level. That is, a downward price movement is expected to most often hit the support level, not pass through it, and then start a climbing price movement. However, should the price actually manage to break through the support level, then the drop is likely to continue until a new support level appears. Conversely, resistance levels define the price that an upward trend is observed to repeatedly bounce against. Likewise, should the price push through this resistance level, then it will continue until another resistance level is encountered. The challenge of achieving the greatest profits using these methods is to find support and resistance levels before they explicitly occur i.e., proactively determine support and resistance. A range of schemes have been proposed for this purpose, including comparisons of price against pivot points (e.g., [132]), moving average envelopes (e.g., [88, 139]), or Bollinger bands (e.g., [88, 23, 24, 33]).

This part of the thesis presents the FXGP component that attempts to utilize retracement levels for the purposes of dynamically characterizing the size of SL and TP orders. SL orders represent an *a priori* rule structured to stem further losses [104]. The converse, take profit orders, act in the predicted direction and result in closing a trading order at a profit. Earlier research assumed that the SL orders were evolved based on the training partition of data [97, 98, 99]. The proposed system with this component successfully evolves trading strategies that both minimize the number of times that a SL order is triggered and minimize the average size of SL orders. To accomplish these goals, it combines the SL orders with the trading rules evolved by FXGP such that they are more proactive and less sensitive to specific thresholds than was previously the case. To do so, this component provides functionality for the three major schemes for SL and TP orders verification and adjustment: Pivot point[2], moving averages, and Fibonacci ratios[3]. The proposed extension of FXGP and it's benchmarking study are detailed in Chapter 5.

2.5 Stock Selection Algorithm for Frequent Intraday Trading (Chapter 7)

Portfolio[4] optimization in the most general of settings attempts to answer the question of how an investor should distribute funds across an available set of investment opportunities.

The pioneering work of Markowitz formulated an approach to portfolio optimization in terms of the following two insights [112, 83]: 1) quantify the return and risk (of specific investments) using statistical estimates of expected return and variance; 2) allocate funds between investments in proportion to the combined return to risk trade-off. These insights also led to the general principle of portfolio diversification.

---

[2] A pivot point is a technical analysis indicator, or calculations, used to determine the overall trend of the market over different time frames. The pivot point itself is simply the average of the high, low and closing prices from the previous trading day.

[3] A Fibonacci Retracement is a popular tool among technical traders. It is based on the key numbers identified by mathematician Leonardo Fibonacci in the 13th century. In technical analysis, a Fibonacci retracement is created by taking two extreme points (usually a major peak and trough) on a stock chart and dividing the vertical distance by the key Fibonacci ratios of 23.6%, 38.2%, 50%, 61.8%, and 100%. Once these levels are identified, horizontal lines are drawn and used to identify possible support and resistance levels.

[4] A portfolio is a grouping of financial assets such as stocks, bonds, commodities, currencies and cash equivalents, as well as their fund counterparts, including mutual, exchange-traded and closed funds.

That is to say, a portfolio's risk is proportional to the correlation *between* the investments. Previous to Markowitz's formulation, the prevailing approach was to instead concentrate on identifying the single asset (stock) that offered the highest future return given the current or historical valuation. Conversely, the approach of Markowitz is based on a 'mean-variance optimization' (MVO) in which a portfolio should be identified as that which has the smallest variance; any other portfolio is considered 'inefficient', i.e. will subject the investor to a higher risk [83].

Since its original formulation, MVO has been widely studied [83]. However, some issues that potentially work against the widespread use of MVO include:

- Estimation errors in forecasted returns and the variance-covariance are poorly behaved [71], where this can result in equally weighted portfolios (i.e. invest in all assets equally) outperforming MVO under practical conditions [38].

- The MVO formulation might result in portfolios that do not reflect the underlying goals of the investors [85]. Alternative formulations such as minimum variance (i.e. ignores the mean returns of an asset, emphasizing instead the correlation between investments) might result in better long term strategies [38]. Likewise, explicitly minimizing the extreme down side investment exposure could capture an underlying investment goal, where the 'conditional value-at-risk' formulation potentially represents an approach for achieving this purpose [138].

- In practice any portfolio is subject to transaction costs which need to be reflected in the MVO formulation. There may also be market specific constraints, such as the ability to short sell (or not), which also need to appear in the investment strategy in order to build a valid portfolio [83].

Another popular stock selection method for long-term investment is the Kelly Criterion [135, 106, 26, 87]. Markowitz portfolios, as shown in [131], do not necessarily minimize risk. However, a Kelly Criterion[5] based portfolio explicitly averages the

---

[5] The Kelly criterion is a mathematical formula relating to the long-term growth of capital developed by John L. Kelly, Jr. The formula was developed by Kelly while working at AT&T's Bell Laboratories. The formula is currently used by gamblers and investors for risk and money management purposes, to determine what percentage of their bankroll/capital should be used in each bet/trade to maximize long-term growth.

long-term return and turns a multi-period optimization problem into single-period optimization problem [131]. The Kelly Criterion does have its own disadvantages, such as assuming that trading activities are performed over a sufficiently long period of time [78]. This issue also limits MVO (see [83], Section 4.3). Conversely, several researchers have also demonstrated that the Kelly criterion can outperform other strategies despite such limitations [13, 122, 131].

In summary, one can note that portfolio analysis is typically a static undertaking applied to asset selection to fund stocks based on fundamental analysis[6] [83] as opposed to, say, stock selection through technical analysis. Indeed, the forecasting of returns and estimation of variance for different assets is difficult enough in settings for which past performance is based on monthly or yearly data. Attempting to apply MVO to frequent intraday trading is significantly harder because the variation of price data at minute intervals is subject to a high degree of variation. Specifically, price data expressed over 10 second intervals [9] (or even 10 minute intervals [108]) has been observed to possess properties such as fat tails (high skewness or kurtosis), where this is particularly undesirable from a risk minimization perspective [9]. These undesirable properties are often referred to as 'microstructure noise'. Approaches assumed for addressing this issue typically take the form of optimizing the sampling windows over which covariance estimates are made. As a consequence the resulting investment models still take the form of weighted combinations of stock from a portfolio.

This work concerns intraday 'frequent trading'. This is distinct from both the high-frequency trading context [59] and the case of asset selection as applied to funding stocks based on fundamental analysis [83]. As were mentioned in Chapter 1, there is still controversy about high-frequency trading (HFT) definitions. One of them characterizes high-frequency trading as relying on the ability to move between short term positions very quickly (microseconds (10^{-6}) or less) at high volumes with profits per trade that might be a fraction of a cent. Research with GP in this context has made recommendations regarding the frequency of trades necessary to improve market stability [109]. Conversely, asset selection based on fund fundamentals we consider a static undertaking, for which the MVO models of Markowitz are well known. Instead

---

[6] Macro or strategic assessment of where a currency should be trading on any criteria but the price action itself. The criteria often include the economic condition of the country that the currency represents, monetary policy, and other "fundamental" elements.

we are interested on the ability to utilize short candlesticks (minutes) within a single trading session (day) (intraday trading) [113, 94].

While there is a growing number of articles covering the development of intraday automated trading systems, the number of works on autonomous asset allocation for such algorithms is limited. Turcas et al. point out the necessity for constructing an optimal portfolio under a frequent trading context [150]. Liu [94] used S&P 500 five-minutes and daily returns to rebalance the portfolio and found that daily rebalancing based on the five-minutes returns gave a performance gain compared to monthly rebalancing.

Ha [61] proposed an intraday trading strategy to absorb the shock to the stock market when an online portfolio selection algorithm is rebalancing a portfolio. The proposed algorithm optimizes both the number of intraday trades and an intraday trading path for a multi-asset portfolio. He used 30 randomly selected NASDAQ 100 components for backtesting and showed that the proposed algorithm was effective for large capital investments and applicable to several portfolio rebalancing strategies.

Borges et al. [161] evaluated the economic benefits of 5, 15, 30, 60, 90 and 120 minutes data for portfolio selection with 30 assets of the Brazilian stock market. They used intraday data (from 5 to 120-minute candlesticks) and compared them with estimators based on daily data. The portfolios were rebalanced daily, weekly and monthly, and are analyzed according to their performance regarding average performance, standard deviation, Sharpe ratio, and turnover. A daily rebalanced portfolio based on the 60 minute returns resulted in a portfolio with lower standard deviation. They concluded that a return sampling frequency ranging from 15 and 120 minutes suggested better performance rather than 5-minute returns. Goumatianos et al. [58] used 1 minute candlesticks of 300 randomly selected stocks from the S&P500 index to build long/short portfolio (10 stocks for long positions and 5 for short).

Common themes that appear in these works include: 1) an emphasis on optimizing the shared temporal window properties over which covariance statistics are estimated; and 2) the use of scalar rebalancing of the investment across the portfolio for the duration of the following trading period. In contrast, this work lets GP trading agents discover trading strategies for each stock. This means that instead of assuming that statistical trends detected in some historical period for a specific stock will just 'carry over' to the next trading period, we actively make use of properties developing within the microstructure of each price signal. This will have implications for both the ranking of stocks (to be traded in the next period), and the ability to make use of price signals within the (intraday) trading period. Other approaches to risk management might include the approach of delta hedging (designing a portfolio such that the value remains unchanged when the value of stock vary), for which a GP approach was recently proposed [160]; and combining both fundamental and technical indicators [32]. However, this work concentrates on quantifying the significance of ranking metrics within portfolios identified by GP trading agents for intraday trading with 1 minute candlesticks. Generally available works let us assume that this is the first time that such an undertaking has been performed.

2.6 On the Effect of Hidden Trading Costs on the Proposed Automated Trading System (Chapter 7)

Trading or transaction costs[7] are one of the most important attributes of any trading system. Elton et al. [51] defined three major sources of trading costs, summarized as follows:

- Commission to the broker, plus any taxes applied to the trade.
- Bid-Ask spread[8], which is defined as a difference in bid and ask prices of a financial asset (stock or a currency pair). This represents a 'roundtrip' cost to the investor buying or selling. The spread can also reflect asymmetries between illiquidity and information [116].
- Potential price impact of a large sale or purchase.

While the first form of transaction cost (commission to the broker plus taxes on the trade) represents explicit (or visible) costs, the other two are implicit (or hidden) costs.

Loistl et al. [103] modeled Xetra[9] and NASDAQ electronic trading systems and compared them in terms of transaction costs (Commission and Bit-Ask spread) for

---

[7] Transaction costs are expenses incurred when buying or selling a good or service.

[8] The distance, usually in pips, between the Bid and Ask prices.

[9] Xetra is the electronic trading system of Frankfurt Stock Exchange.

small (up to 1000 shares), medium-sized and block-sized orders (10000 or more shares). They demonstrated that NASDAQ is much preferable for investors with small orders while Xetra has advantages in case of medium- and block-sized orders.

Duran et al. [48] described a profitable trading and risk management strategy for daily financial decision making. They found that the proposed strategy was profitable for 168 S&R 500[10] and 213 Russell[11] 2000-listed stocks assuming 1% commission for each change of position.

Azencott et al. [8] established a theoretical foundation to a generic framework for real-time market analysis. They describe a methodology for automatically discovering reasons for performance degradation of algorithmic trading, albeit with a focus on transaction costs. However, the quantification of the more variable/hidden sources of trading costs remained elusive.

Li et al. proposed the ‘Transaction Cost Optimization’ framework to improve existing online portfolio selection strategies. They concluded that the proposed framework might effectively address ‘reasonable’ transaction costs. However, the bid-ask spread was beyond the scope of their research [89].

Goumatianos et al. [58] described a stock selection system based on knowledge discovery in large databases. The proposed pattern mining algorithm for time-series was tested by considering only a low fixed commission per trade ($4.95). In short, the hidden costs were not included.

Dempsey et al. [39] reported on the performance of an on-line evolutionary automatic programming methodology for uncovering technical trading rules for the S&P 500 and Nikkei 225 indices, assuming a flat fee model ($10 upon changing of position) with average monthly return as the fitness function. Mabu et al. [105] considered only explicit transaction costs of SBI Securities Co., Ltd which depended on the size of buy or sell order alone. Das et al. [37] introduced an efficient online algorithm for portfolio selection. They demonstrated the effectiveness of the proposed algorithm assuming only proportional transaction costs.

---

[10] The S&P 500 is a stock market index that measures the stock performance of 500 large companies listed on stock exchanges in the United States.

[11] The Russell 2000 Index is a market index of the smallest 2,000 stocks in the Russell 3000 Index. It was started by the Frank Russell Company in 1984.

Several authors have noted the negative impact of bid-ask spread on the performance of funds in general:

Ha and Zhang in [61] proposed an intraday trading strategy to absorb the shock to the stock market when an online portfolio selection algorithm rebalances a portfolio. The results of backtesting from the historical limit order book data of NASDAQ-traded stocks show the effectiveness of the proposed trading algorithm. They considered different types of transaction costs and noted that they excluded trading at the market opening because bid-ask spreads are much higher at that time than mid-day or day closing.

Demsetz in [42] investigated the cost of transacting on the New York Stock Exchange (NYSE) and noted that in case of NYSE two elements comprise almost all the transaction cost — brokerage fees and bid-ask spreads. He characterized the service provided by dealers on NYSE in terms of the 'immediacy' of a transaction. Thus, bid-ask spreads reflect cost per share for supporting the immediacy of a trade. Stoll [143] extended the investigation by introducing a dealer cost function that attributed cost to risk, thus the bid-ask spread reflects a dealer's exposure to risk. Amihud et al. in [4] studied the effect of securities' bid-ask spread on their returns and concluded that market-observed expected return is an increasing and concave function of the spread. In addition, it has been noted that bid-ask spreads can reflect trade execution costs and therefore represent indicators of market quality [14].

Edelen et al. [49] applied an estimate of the trading cost (brokerage commission, bid-ask spread, and price impact) for each position changed and concluded that that hidden trading costs have a detrimental effect on fund performance, where this is at least as material as that of the visible trading costs. Kocinsky [82] noted that the expected value of transaction costs, which takes into account the transaction's volume and duration, may be considered an important measure of a liquidity of a traded stock. Formulations were presented for expected transaction cost, as a function of bid-ask spread and market impact.

Bryant et al. [22] reported on an analysis of spreads before and after the introduction of automation of trading in commodity (futures) markets and noted that spreads have increased since the introduction of automated trading systems. Thus, lower order processing costs in automated trading can be outweighed by increased transaction costs. Moreover, some of the benefits experienced in financial futures markets might not appear under commodity markets as the latter tend to have lower volumes. In addition, other researchers have noted that different commodity markets can exhibit different levels of underlying volatility (e.g. NASDAQ (high) versus NYSE (low)) [73]. The bid-ask spread can also be influenced by the types of traders participating in a market. Informed traders posses information about the true value of an asset, whereas uninformed traders trade for liquidity needs [27]. The potential implication here is that automated trading systems would typically be an example of an 'uninformed' trader, and therefore more susceptible to the negative impact of spreads.

Chapter 7 investigates the impact of the bid-ask spreads on the results of back-testing (and, therefore, the potential impact on the real-time trading) of a genetic programming based automated trading system. Our results concentrate more on the impact of the nature of the spread (fixed or floating) rather than the size of the fixed[12] or floating spreads. A 'fixed spread', in this case, can vary for different trading assets but remain constant throughout the time regardless of the trading asset price or market conditions. Usually, fixed spreads are bigger than the average value of a floating spread but, at the same time, fixed spreads provide more stable trading conditions. This work investigates to what degree (if any) automated trading systems that utilize historical rates to build trading rules for real-time trading may benefit from fixed versus floating spreads.

Early research reported on trading agents identified by genetic programming (GP) with data sourced as spot tick data [41]. However, this data was then converted to midpoint prices that were then aggregated into bars of various frequencies. Unfortunately, this then removes the spread information. Similar limitations appear in [126, 121] where out-of-sample performance of intraday technical trading strategies is selected using two methodologies, a genetic program and an optimized linear forecasting model. Authors of those works concluded that "When realistic transaction costs and trading hours are taken into account, we find no evidence of excess returns to the trading rules derived with either methodology."

Holmberg et al. [65] applied their intraday Open Range Breakout strategy to

---

[12] Bid-Ask spread is fixed when the difference between Bid and Ask prices has a fixed value in pips.

a time series of U.S. crude oil futures prices. They observed that the commission fees and bid-ask spreads will have a negative impact on the profits reported, while the particular nature of the commodity futures market tends to provide a bias towards short-selling. Bitvali et al. [18] performed a trading simulation including fixed transaction costs at 0.6% of the transaction value and concluded that, as expected, transaction costs eroded the profit.

Cirillo et al. [30] assumed tick-level historical price data of four currency pairs and then aggregated this into 5-minutes bars. However, the money management scheme is not detailed and the transaction cost is only reported in terms of the total transaction cost incurred over the evaluation, thus precluding comparative backtesting. Baron et al. [11] investigated risk and return in high frequency trading (HFT) using historical FTSE market data and showed that aggressive HFTs as a whole lose money on shorter time scales; presumably as a result of the bid-ask spread and price impact, although quantification of this did not appear.

Bid and ask prices of two currency pairs (EURUSD and GBPUSD) were used by Mendez et al. [115] to simulate trading using a genetic algorithm. It was concluded that including transaction costs made it significantly more difficult to identify profitable strategies. Vasilakis et al. [153] presented a GP trading technique to forecast the next day returns when trading the EURUSD exchange rate based on historical data. Issues with transaction costs impacting the performance of the GP trading agents again appear with the authors adopting a posteriori 'filtering' approach in an attempt to weed out the more costly trading strategies, an approach also adopted in [110].

Finally, in the special case of executing large institutional orders,[13] genetic programming has been used to design appropriate trade execution strategies [35]. Naturally, such an approach is specific to the case of large institutional orders, taking 5 hours to execute, as opposed to the frequent trading scenario considered in this work.

---

[13] A sufficient number of stocks are traded at a single time to result in moving the prices against their own order.

2.7 Summary

The number of topics investigated in this work and specifics of the application domain make the task of establishing a comparison baseline for the proposed system challenging. Various works, including those mentioned in this Section are designed for different assets and even different markets and tested assuming different trading conditions. As a compromise, this work adopts the following approach. The core of the proposed framework for automating trading agent design consists of a coevolved TI and DT. This will be subject to an ablation study in which the DT is evolved without the benefit of TI, using popular human designed TI, as previously recommended by Hitoshi Iba [68]. This represents the case of evolving trading rules while assuming preprocessed temporal features. The process will be repeated for both Foreign Exchange markets (half hour bars) and intra-day trading on NASDAQ stocks (5 minute bars). At the same time, the proposed process for coevolving TI and DT will be compared to widely used Hoeffding Tree approach for streaming classification. Thus, justifying the use of coevolved TI and DT. The claim is not made that the approach is 'the best', but that the approach to automated trading is competitive with current methods. This will then enable us to concentrate on the impact of transaction costs, asset selection heuristics, and the impact of bid-ask spreads on the automated trading agent.

Chapter 3

Proposed Automated Trading System

3.1 The FXGP Algorithm

The FXGP algorithm is a genetic trading algorithm that utilizes historical prices of a trading asset (currency pair exchange rate, stock price, etc.) to evolve trading agents that then generate real-time trading signals ‘buy’, ‘sell’ or ‘hold’ to open/close a trading position or to wait when the market conditions do not favor trading. FXGP is the core part of the proposed automated trading system. The first version of FXGP (’Base FXGP’ hereafter) was developed as a part of MCS Thesis and was described in [95, 97, 98]. This section summarizes previous results [95, 97, 98] and gives an overview of the Base FXGP algorithm.

3.1.1 Overview of Base FXGP Approach to Constructing Trading Agents

Agents are evolved assuming a ‘Train–Validate–Trade’ cycle (Figure 3.1). Train and Validate represent two sequential historical partitions of the data from which Base FXGP evolves trading agents (DTs with linked TIs) and the best agent is then used for trading. Thus, given *train* sequential records[1] of the Training partition, the TI and DT populations are coevolved. The next *test* sequential records from the Validation partition are used to verify and identify a single champion DT–TI combination (the champion agent). It is possible that the result of model validation is a failure to identify a champion agent, in which case the training cycle would be reinvoked from an entirely new initialization of the DT–TI populations. Assuming that a champion agent is identified then Trading may commence until one of several retrain criteria triggers the identification of a new trading agent. The retrain criteria provide the user with the ability to impart their trading preferences on the operation of the agent (e.g.,

---

[1] Actually expressed as a candlestick tuple $\langle open, high, low, close \rangle$.

conservative versus aggressive). This results in the reinitiating of the 'Train–Validate–Trade' cycle relative to the point at which retraining was triggered (Figure 3.1). Naturally, the Train and Validate partitions are of a fixed duration, whereas the duration of the 'Trade' partition is set by the performance of the champion DT–TI agent. The following subsections will explain each step in more detail.

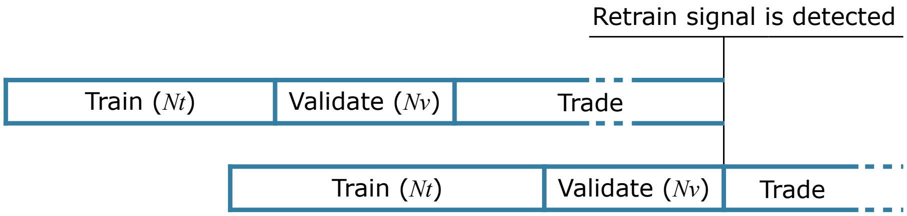


Figure 3.1: The Train–Validate–Trade cycle. Independent populations are evolved during Training partition *train* and validated during Validation partition *test* (adopted from [97])

3.1.2 Training

3.1.2.1 Initializing DT–TI Populations

At the beginning of any training cycle, the population of TI and then the population of DT are initialized.

The *TI population* is randomly initialized with the minimum size defined by the user (Table 3.3). TI individuals assume a linear GP representation (e.g., [21]) with instruction set summarized by Table 3.1. In this case, the genotype is described in terms of a sequence of integers or genes [21]. Each integer value is decoded into a legal instruction and executed on a virtual machine (i.e. the 'fetch-decode-execute' cycle is simulated). Moreover, each TI has a header defining the basic TI properties: links, type, scale, period, shift and length (Table 3.2). However, the size of the TI population is not fixed and can incrementally vary between consecutive generations (Section 3.1.3) but, cannot shrink below the *TIp* (Table 3.3) limit. Moreover, one would desire more flexibility in how TIs are designed, hence rather than choose from a fixed set of TIs or concentrate on parameterizing predefined TI, the explicit evolution of the TIs is preferable. With this in mind the linear GP representation [21, 158] is adopted to generate unique TI with a set of instructions (Table 3.1) [99]. Each TI is characterized by the set of properties stored in a TI header (Table 3.2). The header represents information used to characterize the individual and is later used within the context of selecting individuals for replacement and/or guide application of variation operators (Section 3.1.3).

Function	Definition
Addition	$R[x] \leftarrow R[x] + R[y]$
Subtraction	$R[x] \leftarrow R[x] - R[y]$
Division 1	$R[x] \leftarrow R[x] \div 2$
Division 2	$R[x] \leftarrow R[x] \div R[y]$
Division 3	$R[x] \leftarrow 1 \div R[x]$
Multiplication	$R[x] \leftarrow R[x] \times R[y]$
Square root	$R[x] \leftarrow \sqrt{R[y]}$

Table 3.1: TI functions. $R[x]$ denotes the content of the register x and $R[y]$ denotes the content of either: 1) the register y, 2) a price, or 3) a price TI_{shift} candlesticks back in price history (where $x \neq y$). Note the three forms of division.

Note that "price" here and hereafter can be any one of four candlestick prices: "Open", "High", "Low" or "Close" if not specified explicitly.

Header field	Description
TI_{links}	How many DTs nodes use the TI
TI_{type}	Value, Moving Average (MA) or Weighted Moving Average (WMA)
TI_{scale}	Crosses price or 0 (Figure 3.2)
TI_{period}	MA period (ignored if TI_{type} is Value)
TI_{shift}	Price shift, candlesticks back in a price history
TI_{length}	Number of functions (Table 3.1) in the TI's program

Table 3.2: TI properties. Mutable fields: TI_{type}, TI_{scale}, TI_{period} and TI_{shift}.

The TI program assumes a register level transfer language in which *Regs* (Table 3.3) registers can be addressed. The $R[0]$ register is used as an output register and contains a TI value after program execution. The last appearance of the $R[0]$ as a target register ($R[0] \leftarrow$) in the TI program is defined as TI_{length} and is stored in the TI header.[2]

---

[2] GP does not enforce sequential dependence between instructions, thus code flow does not follow classical properties that human authored code generally possess. Indeed, significant amounts of 'code bloat' (introns) are generally observed [21]. This is a general property of many forms of GP.

Parameter	Description	Default
TIp	Minimum TI population size	100
TIs	Maximum TI program size, steps	6
$Regs$	Number of TI program registers	2
DTp	DT population size	100
$DTgap$	Number of DTs replaced in each generation	25
$DTmut$	Relative probability of DT or TI mutation	0.5
DTs	Maximum DT size, nodes	6
$Gmax$	Maximum number of generations	1000
Nt	Training partition size	1000
Nv	Validation partition size	500
$SLmax$	Maximum SL order size, pips	100
$SLmin$	Minimum SL order size, pips	10
τ	Training plateau length, generations	200
α_v	DT-TI validation fraction	0.9
$Lrow$	Maximum number of consequent losses	3
Dd	Maximum drawdown, pips	400
$Hrow$	Maximum number of consequent candlesticks without trading activity	72

Table 3.3: Base FXGP parameterization.

Figure 3.2: TI scales examples (adopted from [95])

The moving average of a TI (MA) is calculated as (3.1) whereas the WMA type of TI is calculated as (3.2),

$$MA_i = \frac{\sum_{j=n}^{i} V_j}{TI_{period}} \tag{3.1}$$

$$WMA_i = \frac{\sum_{i=1}^{n} \frac{V_i}{i+1}}{\sum_{i=1}^{n} \frac{1}{i+1}} \tag{3.2}$$

where V_i is a TI value and $n = i - TI_{period}$

The TI header fields (Table 3.2) are initialized as follows:

- TI_{links} is set to 0.
- TI_{type} is randomly initialized with 0 (Value), 1 (MA) or 2 (WMA).
- TI_{period} is randomly initialized with an integer that satisfies the following condition: $1 < TI_{period} \leqslant period$ or ignored if TI_{type} is Value.
- TI_{shift} is randomly initialized with an integer that satisfies the following condition: $0 \leqslant TI_{shift} \leqslant shift$.

The TI_{scale} is detected after initialization[3] and then checked following application of the mutation operator in order to detect a degenerate[4] TI. If the scale is not valid, the TI is discarded and initialization is repeated until the scale condition is satisfied. Figure 3.3 shows the distribution of TI programs' length (instructions) after the cycle of training which indicates that TI typically consist of between 2 and 5 instructions. Hence, although 'short' they are not merely referencing single price points, but constructing a temporal feature. (See also program examples in Appendix B).

Following initialization of the TI population, the DT population size of *DTp* (Table 3.3) is initialized with *tree* individuals. The DT population assumes a tree structured representation describing what will become 'trading rules'. Such rules represent antecedent–consequent pairs that GP uses to define trading decisions resulting in one of three outcomes: $\{buy, hold, sell\}$. Each DT is characterized by a set of properties stored in the DT header (Table 3.4). Moreover, each DT can vary in size, however, the

---

[3] A TI with positive and negative values is said to be zero crossing 0 imply that $TI_{scale} = 0$. A TI with values crossing the stock's price value imply that $TI_{scale} = 1$

[4] "Degenerate" TI is a TI that does not cross the price or 0.

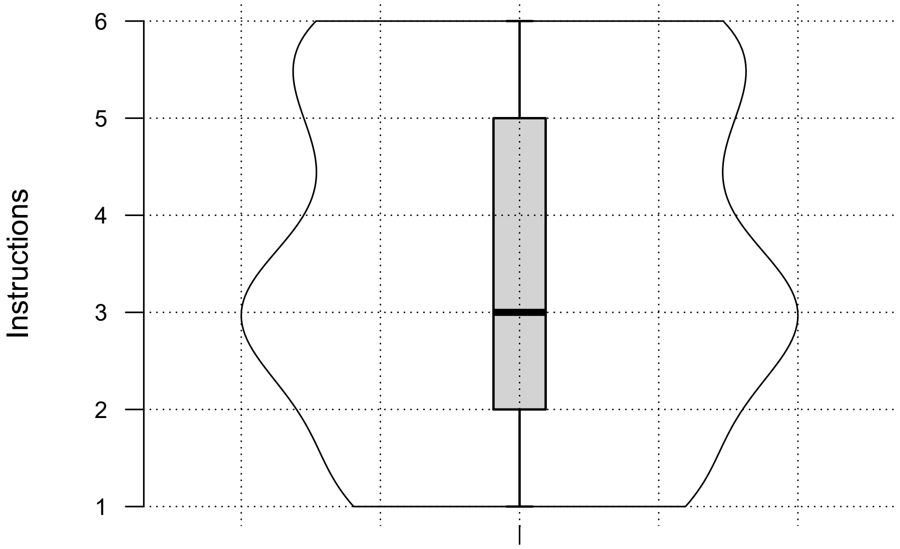


Figure 3.3: TI programs' length (instructions) distribution after the cycle of training.

number of nodes is limited as follows: $1 < DT_{size} \leqslant nodes$. The DT_{score} is measured in pips.

Base FXGP makes use of Stop-Loss (or SL) orders to limit a loss when the price is moving in the opposite to the desired direction. A SL order can be a fixed size (fixed SL) or can be evolved (floating SL). In the last case, its size can vary between SL_{min} and SL_{max} (Table 3.3). The actual size of the SL order is stored in the DT header (DT_{sl} field). In case of floating SL, the DT_{sl} is initialized with a randomly selected value between the minimum (SL_{min}) and maximum (SL_{max}) SL order sizes.

Header field	Description
DT_{size}	DT size, nodes
DT_{score}	DT score, pips
DT_{trades}	Number of trades
DT_{sl}	SL order price, pips

Table 3.4: DT properties. DT_{sl} is the only mutable fields.

Each DT node is represented as one of the following conditional statements [99]:

- $IF\ (X_i > Y_i)\ THEN\ \langle arg_1 \rangle\ ELSE\ \langle arg_2 \rangle$
- $IF\ ((X_i > Y_i)\ and\ (X_{i-m} < Y_{i-m}))\ THEN\ \langle arg_1 \rangle\ ELSE\ \langle arg_2 \rangle$

where m is a time shift (candlesticks) back in a price history, $X_i \neq Y_i$ and X_i and Y_i are randomly selected and can be: 1) zero, 2) a price, or 3) a TI. However, in order to enforce meaningful comparisons the following restrictions are enforced:

1. if X_i is 0, then Y_i must be a TI that crosses 0.
2. if X_i is a TI that crosses 0, then Y_i must also be a TI that crosses 0.
3. if X_i is a price or a TI that crosses the price, then Y_i must also be a price or a TI that crosses the price.

Likewise, arg_1 and arg_2 are randomly assigned as either a pointer to a next node, or one of the following trading signals: $\{buy, hold, sell\}$ subject to the constraint that $arg_1 \neq arg_2$. Note also that additional TI can be generated during the DT population initialization if there are no TI in the TI population that satisfy the restrictions of the above three conditions.

Poli at al. in [133] mentioned that *grow* and *full* are two simplest methods of tree initialization. Their combination is known as *Ramped half-and-half* [84]. In all cases, trees are initialized to not exceed the user-specified depth. The *full* method generates trees with all leaves at the same depth, while the *grow* method results in trees of varied sizes and shapes. At the same time, the *grow* method is highly sensitive to the sizes of function and terminal sets. If the number of terminals is greater than the number of functions, the *grow* method will generate short trees. Otherwise, it will behave almost like the *full* method. This work relies on the *grow* method to allow trees of various shapes and sizes. In this case, each node represents a single trading rule in its simplest form. The trees' size limited not by the depth but by the total number of nodes or rules. To overcome the dependence of the *grow* method sensitivity to the sizes of function and terminal sets, the type of the branch (function or terminal) is randomly selected with the uniform probability and then the function or the terminal is randomly chosen (uniform probability) from the corresponding set. DT population statistics collected after a single initialization are shown on Figures 3.4 and 3.5.

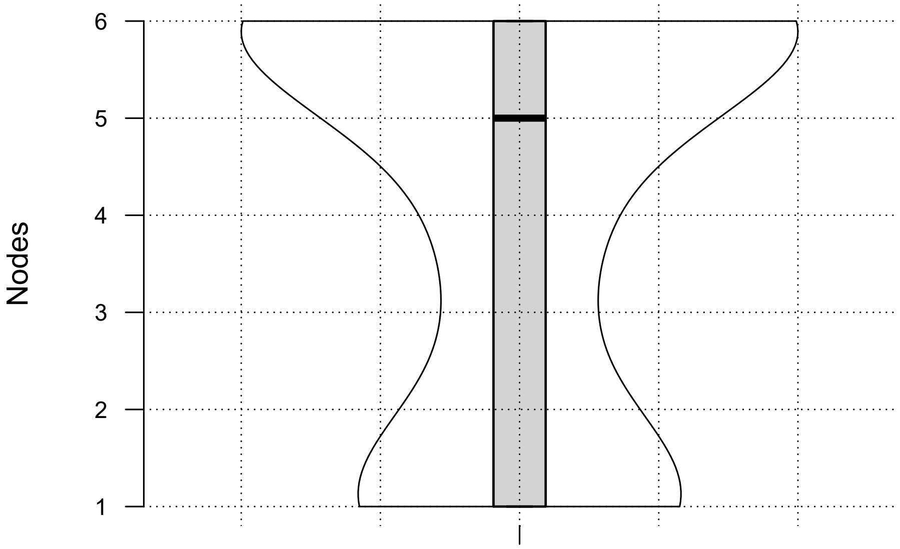


Figure 3.4: DT size distribution over a single population of 100 trees.

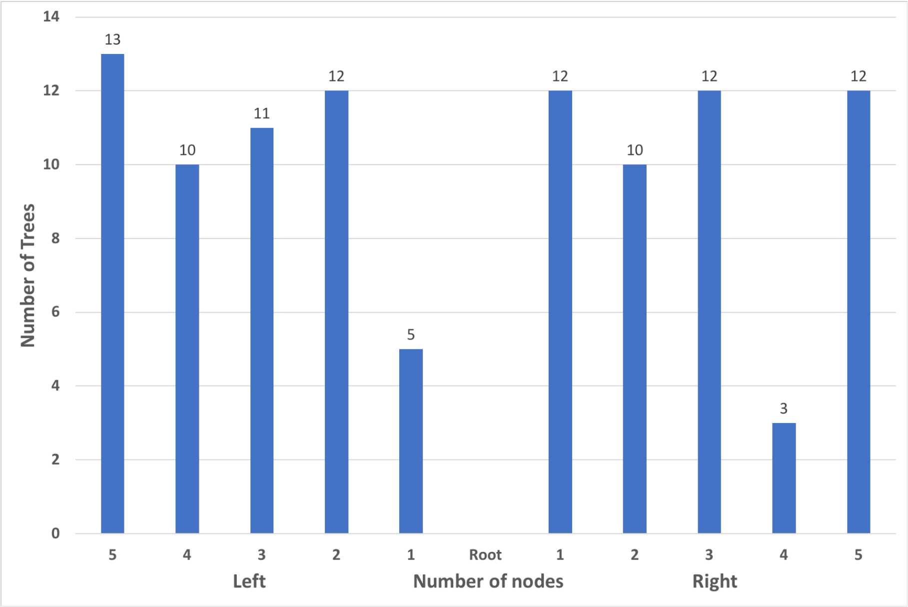


Figure 3.5: DT shape (number of nodes in the left and right parts) distribution over a single population of 100 trees.)

3.1.3 Selection and Variation

At least $DTgap$ (Table 3.3) DT individuals are replaced per generation. DT individuals are selected for replacement based on statistics collected in the header fields of DT individuals when they are evaluated (Table 3.4). Specifically, the DT_{trades} and DT_{score} fields are used to prioritize DT individuals for replacement during a two stage selection process:

Stage 1: The first n DT individuals with $DT_{trades} == 0$ are selected for replacement.

Stage 2: IF $n < DTgap$ THEN the next $DTgap - n$ trees with the lowest DT_{score} are selected for replacement.

After identifying the $DTgap$ DT for replacement, the corresponding TI used by these DT have their TI_{links} counter decremented. Any TI with a $TI_{links} == 0$ are considered useless and deleted resulting in a decrease in the size of the TI population by a variable amount. The DT population in contrast always looses (and gains) $DTgap$ individuals at each generation.

Variation operators modify the genotype of parents to define new candidate solutions or 'offspring' (Section 2.1). Variation operators exchange sequences of the genotype between two parent individuals (crossover) or modify specific genes (mutation). Hence, crossover creates a new individual by mixing material that currently exists in the population, while mutation can introduce new genetic material to the population. Chellapilla in [28] empirically demonstrated that crossover does not necessarily lead to better results than mutation on a suite of 14 benchmarks. Angeline in [6] argued that subtree crossover is the main cause of bloat if compared to subtree mutation. Moreover, Banzhaf et al. [10] demonstrated the positive effect of extensive use of the mutation on generalization in GP using sparse data sets. Considering the above and performance reasons, this work utilizes the mutation as the only variation operator.

Following replacement, mutation is used to add $DTgap$ DT offspring. Parent DT are selected randomly among the remaining $trees - DTgap$ individuals. A two stage process is assumed in which a parent is first cloned and a target for the mutation operation identified as either the DT or a linked TI. The $DTmut$ (Table 3.3) probability sets the relative probability of mutating a DT or (linked) TI. Only one DT node or one (linked) TI can be mutated. If the mutation target is a DT node, one of the following mutation operations is applied:

- New conditional function.
- New shift parameter m.
- New X_i.
- New Y_i.
- Interchange X_i and Y_i within the same node.
- Switch content of *then* and *else* clauses.
- New *then* clause content.
- New *else* clause content.
- New SL order size.

If the mutation target is a TI, the parent TI is cloned replacing an existing TI with $TI_{links} == 0$ or creating a new TI (implying an increase in the size of the TI population) and then one of the following mutation operators is applied:

- New TI_{type} ($Value,\ MA\ or\ WMA$).
- New TI_{period} (if TI_{type} is MA or WMA).
- New TI_{shift}.
- Generate a new instruction (Table 3.1).
- Delete an instruction (if $TI_{length} > 1$).
- Insert an instruction (if $TI_{length} < length$).

In both cases (DT and TI mutation), if the mutable parameter can take more than two values, the new value is randomly chosen with uniform probability. Training stops when the maximum number of generations (*generations*) is reached or when the best DT score remains the same for specified number (*plateau*) of consecutive generations.

3.1.4 Validation

As highlighted in Section 3.1.1, a Validation partition, containing price data independent from the Training partition (Figure 3.1), is used to verify the quality of the whole DT–TI population and then, if the DT–TI population passes the test, identify the DT–TI agent with the best DT_{score}. The latter step establishes a single champion DT–TI individual that will then be used as the trading agent.

The DT–TI population quality control is performed in order to verify performance. Specifically, the subset of DT–TI individuals that are within $\alpha_v = 90\%$ of the best DT–TI after the training is identified. This subset of individuals is then evaluated on the Validation partition to identify the the champion. With this in mind, validation has the following form:

1. Compare the scores of all trading agents on the Training partition against the best agent's score DT^{best}_{score};
    - (a) IF $DT_{score} > \alpha_v \times DT^{best}_{score}$ THEN reset DT header fields DT_{score} and DT_{trades} to 0;
    - (b) Apply the DT–TI agent over the *test* data from the Validation partition (Figure 3.1);
    - (c) IF $DT_{trades} > 0$ after Validation THEN increment the DT agent's counter n (Algorithm 1, line 11);
2. IF $n \geqslant (trees - gap)$ after validation THEN select DT–TI agent with the highest score for trading (Algorithm 1, lines 12 to 16);
3. ELSE discard the DT and TI populations and repeat training on Training partition (Algorithm 1, line 4).

3.1.5 Trading

The next component of Base FXGP answers the issue of how long the current DT–TI champion(s) should be deployed (Figure 3.1). Base FXGP continuously monitors the trading performance of the trading agent(s) (Algorithm 1, lines 16. . . 18) against which a set of change detection criteria (Table 3.3) are deployed as follows:

Algorithm 1: The core Base FXGP algorithm. *Retrain* refers to one of the retrain criterion

```
Input: The historical FOREX or Stock data
Output: Trading rule (DT with linked TIs)
$t = first\ candlestick\ of\ the\ first\ day$        // define start of trading
while $t \leqslant last\ candlestick\ of\ the\ last\ day$ do
    $best = NULL$        // reset DT-TIs champion
    while $best == NULL$ do
        initialize TI population
        initialize DT population
        evolve DT and TI populations over ($train$)
        $n = 0$        // DT-TIs agent count
        for $i = 0\ to\ trees$ do
            if $(validate\ DT_i\ over\ test) == TRUE$ then
                $n++$
        if $n >= (trees - DT_{gap})$ then
            for $i = 0\ to\ n$ do
                find (DT-TIs)$_i$ agent with the highest score
            $best = i$        // update DT-TIs champion
    while $Retrain == False$ do
        trade
        $t++$
```

- *Dd* — the maximum account balance drawdown
- *Hrow* — the maximum period of time (price candlesticks) without trading activity
- *Lrow* — the maximum number of consecutive losses

When any one of these criteria is satisfied, Base FXGP stops trading and initiates a new Train–Validate cycle to build a new multi-agent team. Such an approach is adopted on account of the inherently non-stationary nature of price data on which trading decisions are made. A previous study confirming that explicitly trapping and then retraining was preferable than attempting to continuously incrementally train Base FXGP individuals [97]. Naturally, the above criteria are a reflection of the particular preferences of the user, without a claim of their optimality. Indeed, users would be free to assume any number of parameterizations or combinations of criteria for the purpose of detecting changes to the underlying price data.

3.1.6 Performance Evaluation

3.1.6.1 FOREX Historical Rates

To evaluate the performance of the Base FXGP algorithm historical rates of the three major currency pairs were used: EURUSD, USDCHF, and EURCHF. The one-hour candlestick historical rates were obtained from the Forex Historical Data repository.[5]) over the trading period from January 2, 2009 to December 30, 2011 (approx. 18,500 hours). These pairs are actively traded on all markets around the world i.e., a trading day consists of 24 one-hour candlesticks (except on weekends and holidays). Sampling at the rate of one-hour intervals was chosen to reduce the impact of "trading noise" [70]. The Skewness and Kurtosis metrics of all three currency pairs ('Close' prices) are shown in the Table 3.5.

Currency pair	Skewness	Kurtosis
EURUSD	-0.29507	2.63075
USDCHF	-0.41103	2.43596
EURCHF	-0.08153	1.87149

Table 3.5: The Skewness and Kurtosis metrics of currency pairs

Skewness is a dimensionless descriptive statistic that characterizes the degree of asymmetry present in a distribution. Positive Skewness characterizes the direction and relative magnitude of deviation from the normal distribution. Thus, positive (negative) Skewness implies that more than half the distribution lies to the left (right) of the mean, but the majority of the largest deviations are to the right (left) of the mean [128]. Kurtosis characterizes the frequency of extreme deviations (outliers) from

---

[5] `http://fxhistoricaldata.com`

the mean. In the case of univariate normal distributions, a Kurtosis less than 3 implies that there are proportionally less outliers than experienced in a normal distribution. Conversely, as the Kurtosis increases beyond a value of 3, the frequency of outliers increases (w.r.t. those expected from a normal distribution).[6] Table 3.5 therefore indicates that for this particular 3 year trading period there is an underlying negative trend (reflected in the Skewness), and the EURCHF currency pair had considerably less Skewness and outlier frequency than EURUSD and USDCHF.

3.1.6.2 Evaluation of Specific Base FXGP Features

Experimental setup. Specific Base FXGP features are evaluated using historical rates of the most widely traded currency pair EURUSD using 1-hour candlesticks, Section 3.1.6.1. The historical rates cover the three year period July 2009 to November 2012 (or $\approx 17,860$ hrs) and is described in terms of the following fields: Pair, Date, Time, bid price Open (Open), bid price Low (Low), bid price High (High) and bid price Close (Close). For evaluation purposes, the *fixed spread* value of $0.00002 was assumed. The spread value was defined based on the FxPro Group average EURUSD spread value.[7] Specifically, the relative contribution of the following parts of Base FXGP was evaluated:

- Criteria based re-triggering of the retraining event (Section 3.1.5) versus the widely assumed case of continuously evolved trading agents as described below.

- Significance of including a validation partition. Base FXGP assumes that the champion individual is identified by a validation partition. However, this also implies that the most recent data is not available for training. The inclusion and non-inclusion of the validation partition will also be considered.

- Degree of support for evolving market specific thresholds. This case considers two different forms of SL thresholds (Section 3.1.2.1). Either a fixed threshold ($DT_{sl} = 100$ pips) or thresholds evolved over a market specific spread interval ($DT_{sl} = 5 \dots 100$ pips) (Table 3.4).

---

[6] `https://www.jstor.org/topic/kurtosis/`

[7] http://www.fxpro.co.uk

Framework for continuous re-evolution. Base FXGP adopts a retriggering scheme for determining when to deploy a new trading agent (Section 3.1.5). However, we also want to understand the significance of assuming continuous models of evolution for evolving trading agents e.g., [40, 68]. To do so, the 'rolling window' approach to evolution was assumed [68]. The DT–TI trading agents are evolved in the same way as Base FXGP (over the training and validation partitions N_t and N_v respectively). Then the champion trading agent is deployed and trading is performed over a fixed number of candlesticks (trading partition N_{trade}) (Figure 3.6). Also, in the case of an open trade by the end of the period, the position will not be closed until the closing signal is generated, and that can result in an extended trading partition. After the end of a trading period N_{trade}, the training and validation partitions are realigned and retraining is restarted (Figure 3.6).

This means that the existing DT–TI populations are either:

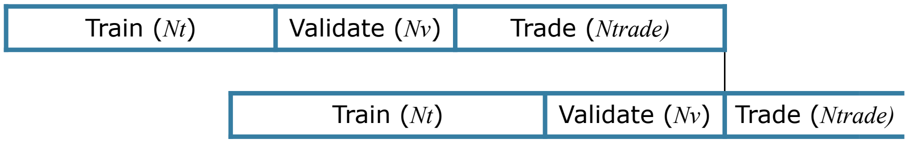


Figure 3.6: Identifying a new champion trading agent through rolling window. The Train–Validate–Trade cycle. Independent populations are evolved during Training partition *train* and validated during Validation partition *test*. Trading is performed for a fixed number of cycles (N_{trade}) before a new cycle of evolution is performed (adopted from [97])

- Used to continue evolution. This approach is adopted by modern researchers (e.g., [40, 68]) and will be referred as continuous evolution or **ContEv**. This naturally leads to exploiting previously evolved rules and makes impossible to use the validation criteria from Section 3.1.4.

- Discarded and new DT and TI populations are initialized. Hence, any previously evolved DT–TI trading agents are replaced with new ones after the first cycle of evolution. This approach emphasizes exploration and is similar to the approach adopted in [158], and will be referred to as stepwise evolution or **StepEv**. Also, this case allows the use of a validation partition.

In total, four cases were evaluated: Base FXGP, StepEv (stepwise evolution), ContEv (continuous evolution; representing current practice) and Static. The Static represented a baseline DT–TI model in which no retraining is performed after the initial cycle of training and just one trading agent is used over the whole period of time from January 2, 2010 to November 30, 2012.

Results. Figure 3.7 provides a high-level summary for all four cases. The returns were collected over 100 runs in each case.

The default parameterization (Table 3.3) was used during the experiments. The default values of the parameters were determined[8] via experiments performed in previous works [98] and [97].

In short, the following general trends were apparent at that stage [95]:

- Retaining population content between cycles of retraining (ContEv) appears to be detrimental to providing new trading agents capable of reacting to the next cycle of trading. However, neither ContEv nor StepEv appear to match the effectiveness of Base FXGP (dynamically identifying retraining intervals).

- Both Base FXGP and StepEv (the latter for trading periods of 1 to 4 weeks) have a strong preference for including the validation partition. Interestingly for the longer trading period under StepEv ($N_{trade} = 1500$) there was no benefit in including a validation partition.

- Enabling the evolution of SL thresholds did not result in a general pattern of preference for or against its inclusion. However, it does appear to be useful w.r.t. ordering Base FXGP.

- In the case of the lengths of trading considered with StepEv and ContEv — $N_{trade} \in \{120, 500, 1500\}$ — the longest trading period was most effective with ContEv, but least effective with StepEv. Further research would be necessary to obtain a clearer picture of the dynamics behind this.

- It is possible that any configuration can result in profitable (unprofitable) runs. However, in most cases at least 25% of runs are unprofitable. Given that it is

---

[8] No claims regarding optimality of the parameters are made.

not a priori possible to determine which runs will be profitable over the investment period, this means that it is very important to characterize performance over multiple runs. If only best-case runs are considered, then even the static configuration is ‘profitable’.

- Training once at the beginning of the three year period and then assuming that the resulting champion individual will be effective clearly does not work. This illustrates that the FX trading task is essentially non-stationary. Naturally, the use of a validation partition is irrelevant as 1 month of validation data is not sufficient for characterizing the following three years of trading when the underlying process is non-stationary.

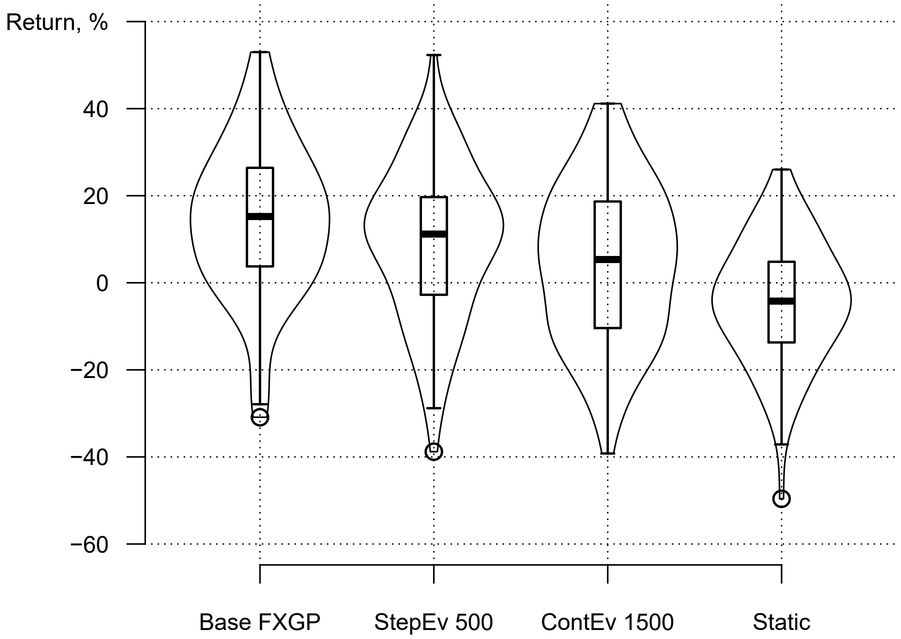


Figure 3.7: Comparison of distribution of returns for Base FXGP, StepEv, ContEv and Static cases. 100 runs per distribution

3.2 Base FXGP Critique

While the above results have provided an initial ‘proof-of-concept’ for the effectiveness of the Base FXGP algorithm, drawbacks of the Base FXGP approach can be summarized as follows:

1. $R[x]$ and $R[y]$ can be negative or ≈ 0. Therefore, two division functions ($R[x] \leftarrow R[x] \div R[y]$ and $R[x] \leftarrow 1 \div R[x]$) and a square root $R[x] \leftarrow \sqrt{R[y]}$ (Table 3.1) require additional checks and decisions to be made in case of illegal operations. This work does not utilize the protected operators [84] since any “adjustment” of a TI value can result in a bad trading decision and, therefore, can lead to a significant loss.

2. Each TI_{scale} TI’s header field requires that a scale check be performed and mark a TIs as ‘bad’ if the scale does not satisfy requirements.

3. Calculation of WMA (3.2) requires much more computational time than MA (3.2), especially for big WMA or MA periods TI_{period} but at the same time, the advantage of using WMA over MA is not clear.

4. The size of DTs tends to grow and reaches the limit (DT_s) within the first generations. This can be explained by the growing number of intron nodes in the DT.

5. The proposed framework supports only one type of limit orders — SL orders. However, the introduction of Take-Profit orders (TP) can help to fix the profit when the price reaches a desirable level.

6. There are three tests applied to assess the validity of the DT (end of Section 3.1.2). Unfortunately, there is currently no guarantee that children will satisfy these tests. It would be much more efficient if DT could be constructed in such a way so as to always lead to valid tests.

7. Lack of trustability to deployed trading agents. While FXGP demonstrates the high median profitability over 100 simulations, the distribution of results due to initial conditions is undesirable and lies in the range from -30% loss to more

than 50% profit (Figure 3.7, ‘Base FXGP’). Given that a trader would not be in a position to know a priori which runs to ignore, identifying a scheme for increasing the certainty of results would be beneficial.

The first four mentioned drawbacks result in the growth of computational overhead, and their elimination is essential for developing a real-time trading system. Improving the functionality to TP orders would improve the algorithm’s performance without increasing complexity. Likewise, addressing drawback 6 would also improve the efficiency of the search process, whereas addressing the last drawback would significantly improve the usability of an autonomous trading agent.

3.3 The FXGP Algorithm, Revised

The issues summarized in Section 3.2 prompted a review of the Base FXGP algorithm (Section 3.1) resulting in this thesis proposing the following improvements:

In the case of **issue 1** the original TI function contains seven instruction types (Table 3.1), whereas analysis of the programs of the resulting TIs indicated that only three functions where typically employed (Table 3.6). Thus, multiplication that produces a TI with the wrong scale, two division functions and a square root function that frequently resulted in illegal operations (e.g. division by zero or square root of negative value) were all dropped from the instruction set. Needless to say, this also removes instruction types that have a longer (computational) latency i.e., division and square root; thus an expected improvement to TI execution, where it is the evaluation of TI that account for the majority of CPU time. Table 3.6 summarizes the updated set of TI functions.

Function	Definition
Addition	$R[x] \leftarrow R[x] + R[y]$
Subtraction	$R[x] \leftarrow R[x] - R[y]$
Division	$R[x] \leftarrow R[x] \div 2$

Table 3.6: The updated set of the TI functions. $R[x]$ denotes the content of the register x and $R[y]$ denotes the content of either: 1) the register y, 2) a price, or 3) a price TI_{shift} candlesticks back in price history (where $x \neq y$ and x and $y \in \{0, 1\}$).

In the case of **issue 3** there was originally three types of TI were supported:

Value, MA and WMA 3.1. This will be revised to support only two types: Value and MA (Table 3.6). The calculation of the WMA requires more computational resources compared to estimation of the MA (Section 3.2). At the same time experimentation indicated that the effectiveness of the WMA failed to improve on that using TI based on MA alone, resulting in the simplified set of TI parameters. In short, supporting more complex moving average definitions does not appear to have any functional benefits.

In the case of **issue 2**, the TI_{scale} parameter was dropped, along with the introduction of DT intron node detection and removal as described below. Table 3.6 summarizes the updated set of the TI properties.

Header field	Description
TI_{links}	How many DT nodes use the TI
TI_{type}	Value original Moving Average (MA)
TI_{period}	MA period (ignored if TI_{type} is Value)
TI_{shift}	Price shift, candlesticks back in a price history
TI_{length}	Number of functions (Table 3.1) in the TI's program

Table 3.7: The updated set of the TI properties.

To implement DT intron node detection and removal two more fields were added to each node: N_{then} and N_{else} transition counters. The N_{then} and N_{else} counters of each node are set to '0' when the new DT is initialized, i.e. generation of the initial DT population or after mutation of a parent DT. During fitness evaluation (Section 3.1.2), if any of the conditional statements is satisfied, the corresponding counter (N_{then} or N_{else}) is incremented. As a consequence, all DT nodes with either transition counter (N_{then} or N_{else}) equal to '0' are considered introns and removed. The TI_{links} (Table 3.7) counters of the corresponding TIs are decremented, so removing all redundant TIs (e.g. TIs with invalid scales, etc.). In summary, dropping the 'TIscale' parameter makes redundant the enforcement of conditional statements as listed at the end of Section 3.1.2 (**issue 6**), whereas the introduction of DT intron node detection and removal resolves **issue 4** from Section 3.2.

To address the **issue 5**, Section 3.2, the TP order field was added to the DT header (Table 3.8) and two more parameters (TP_{max} and TP_{min}) were added to the base FXGP parameterization (Table 3.9).

Header field	Description
DT_{size}	DT size, nodes
DT_{score}	DT score, pips
DT_{trades}	Number of trades
DT_{sl}	SL order price, pips
DT_{tp}	TP order price

Table 3.8: The updated set of the DT properties.

Parameter	Description
TIp	Minimum TI population size
TIs	Maximum TI program size, steps
$Regs$	Number of TI program registers
DTp	DT population size
$DTgap$	Number of DTs replaced in each generation
$DTmut$	Relative probability of DT or TI mutation
DTs	Maximum DT size, nodes
$Gmax$	Maximum number of generations
Nt	Training partition size
Nv	Validation partition size
$SLmax$	Maximum SL order size, pips
$SLmin$	Minimum SL order size, pips
$TPmax$	Maximum TP order size, pips
$TPmin$	Minimum TP order size, pips
τ	Training plateau length, generations
α_v	DT-TI validation fraction
$Lrow$	Maximum number of consequent losses
Dd	Maximum drawdown, pips
$Hrow$	Maximum number of consequent candlesticks without trading activity

Table 3.9: The updated FXGP parameterization.

The set of the mutation operation (Section 3.1.4) was updated according to all changes. If the mutation target is a DT node, one of the following mutation operations is applied:

- New conditional function.
- New shift parameter m.
- New X_i.

- New Y_i.
- Interchange X_i and Y_i within the same individual.
- Switch content of $then$ and $else$ clauses.
- New $then$ clause content.
- New $else$ clause content.
- New SL order size.
- New TP order size.

If the mutation target is a TI, the parent TI is cloned replacing an existing TI with $TI_{links} == 0$ or creating a new TI (implying an increase in the size of the TI population) and then one of the following mutation operators is applied:

- New TI_{type} ($Value$ or MA).
- New TI_{period} (if TI_{type} is MA).
- New TI_{shift}.
- Generate a new function.
- Delete a function (if $TI_{length} > 1$).
- Insert a function (if $TI_{length} < length$).

Finally, with regards to **issue 6**, the set of conditional statements (Section 3.1.2) was updated. Each DT node is now represented as one of the following conditional statements [99]:

- $IF\ (X_i > Y_i)\ THEN\ \langle arg_1 \rangle\ ELSE\ \langle arg_2 \rangle$
- $IF\ (X_{i-m} < Y_{i-m})\ THEN\ \langle arg_1 \rangle\ ELSE\ \langle arg_2 \rangle$

where m is a time shift (candlesticks) back in a price history, $X_i \neq Y_i$ and X_i and Y_i are randomly selected and can be: a price or a TI. The meaning of arg_1 and arg_2 was left unchanged.

The updated set of conditional statements improved the performance of the proposed FXGP algorithm, made the DT easier to decode and analyze, and has improved the flexibility of the DT by increasing the number of multi-conditional (chained) statements the algorithm can build.

3.3.1 Multi-Agent FXGP

This section describes the new addition to the proposed algorithm, which allows the use of multiple trading agents ('Teams' further on) to suggest the trading action (hereafter in this section FXGPT). As will be shown below, the simultaneous use of multiple trading agents (teams) helps to improve the median profit and reduce the sensitivity of results due to initial conditions (issue 7, Section 3.2).

3.3.1.1 Constructing FXGP Teams

Multiple FXGP populations are to be evolved relative to the current historical trading data. That is to say, each time the retraining criteria flags poor trading behavior, all populations will be re-evolved.[9] The team is built in two ways:

Mode 0: Given P independent FXGP populations, identify one champion trading agent from each using the validation data, N_v (Figure 3.8).

Mode 1: As per mode 0, however, all FXGP individuals passing the validation criteria form the basis for a new population, p^*. This population continues evolution with respect to partition N_{team} (Figure 3.8). Note that each individual from p^* is still treated as an independent trading agent.

Post evolution, each trading agent in the team returns one of three actions per trading interval (hourly in this section), where actions are mapped to an integer value using the following assignment: Sell = -1; Stay = 0; Buy = 1. The scheme adopted for combining the recommendation from each agent assumes the following form: $a = \sum_{i \in A} a_i$ where $a_i \in \{-1, 0, 1\}$ corresponds to the three possible actions

---

[9] FXGP utilized three criteria: 1) max. single drawdown, 2) max. number of consecutive loss making trades, 2) maximum number of candlesticks without trading activity 3.1.

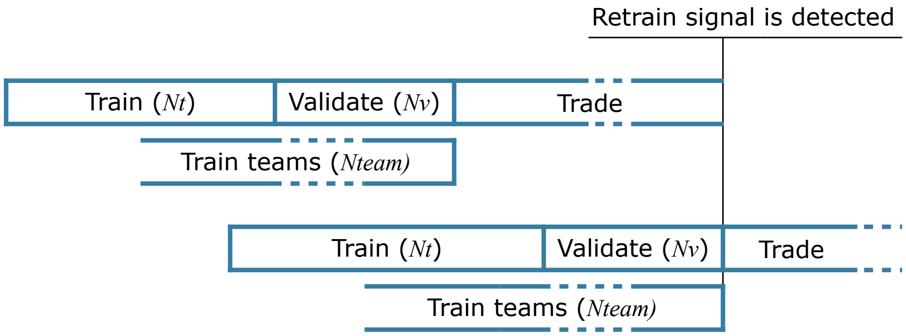


Figure 3.8: The Train–Validate–Train–Trade cycle. Independent populations are evolved during Training partition N_t and a subset of sFXGP individual identified from each using the Validation partition N_v. The FXGP teams are trained over partition N_{team} (mode 1 only). The point at which retraining is invoked corresponds to three trading criteria.

that each champion can assume and A is the strongest subset of agents from p^* at the last generation. The resulting number line is then re-mapped into one of the three actions using the following rule (3.3):

$$\text{IF } (a \geq b) \text{ THEN } (buy) \text{ ELSE IF } (a \leq -b) \text{ THEN } (sell) \text{ ELSE } (hold) \tag{3.3}$$

Naturally, the value for the threshold b needs to be defined by the user and remains the same throughout the trading activity. The use of the user-defined threshold instead of simple voting allows users to implement a ‘veto’ mechanism for each trading agent on a team by adjusting the threshold value. Also, noted that the generic form of this model has been adopted in the past for discretizing the output of (single) neural network trading agents into long and short positions [47] and ‘risk management’ in the case of boosted DT (γ_0 parameter in [34]).

The introduction of teams adds to the base FXGP parameters (Table 3.3) a set of team-specific parameters as listed in the Table 3.10.

The updated structure of the multi-agent FXGPT algorithm is shown below (Algorithm 2):

3.3.2 Validation of the Updated Algorithm

This section pursues two main goals:

Parameter	Description	Value†
$Tsize$	Number of trading agents in a team (number of independent DT–TI populations)	3
b	Team's trading signal threshold (3.3)	2
$mode$	0 - team is built with champion agents 1 - team is evolved	0 or 1
Tp	Teams' population size ($\|p^*\|$)	100
$Tgap$	Number of teams to be replaced per generation	25
$TGmax$	Max number of teams' population generations	1000
T_τ	Team training plateau length, generations	200
Tt	The data partition size to evolve teams	500

Table 3.10: Team specific parameters. Value† represents the parameterization used to perform experiments in this section.

1. Validate the effectiveness of the introduced changes and assure that they do not reduce the performance of the algorithm.
2. Investigate the influence of different trading conditions, namely the fixed and floating Bid-Ask spreads, on the results of the trading simulation and, therefore, on the results of real-time trading.
3. Validate the effectiveness of the multi-agent approach and define the best way to build a team of trading agents (utilize the best agents in each DT-TI population or evolve the team over the N_{team} partition of data).

The EURUSD tick-by-tick prices[10] were converted into one hour candlesticks and used to define market activity during the period from January 3, 2010 to November 30, 2012. To achieve the first goal and establish a baseline for comparison, the same period of time as in Section 3.1.6.2 is used. The results in Section 3.1.6.2 were obtained with the assumption of a *fixed spread* value of 0.00002 USD based on the FxPro Group average EURUSD spread value.[11]. Therefore, to achieve the second goal, the downloaded historical rates include the real floating Bid-Ask spreads. The distribution of floating spreads (the difference between Ask and Bid prices) of the hour candlesticks during trading (Open, High, Low and Close prices) are shown in the Figures 3.9 and 3.10.

---

[10] http://www.truefx.com

[11] http://www.fxpro.co.uk

Algorithm 2: The multi-agent FXGPT algorithm. *Retrain* refers to one of the retrain criterion

```
Input: The historical FOREX or Stock data
Output: Trading rule ($T_{size}$ DTs with linked TIs)
$t = first\ candlestick\ of\ the\ first\ day$          // define start of trading
while $t \leqslant last\ candlestick\ of\ the\ last\ day$ do
    $teams = NULL$                                    // no DT-TIs agents identified
    for $k = 0$ to $T_{size}$ do
        $best = NULL$                                 // reset DT-TIs champion
        while $best == NULL$ do
            initialize TI population
            initialize DT population
            evolve DT and TI populations over ($train$) detecting and removing
              intron DT nodes in each generation
            $n = 0$                                   // DT-TIs agent count
            for $i = 0$ to $trees$ do
                if (validate $DT_i$ over test) == $TRUE$ then
                    $n{+}{+}$
            if $n >=$ ($trees - gap$) then
                for $i = 0$ to $n$ do
                    find (DT-TIs)$_i$ agent with the highest score
                $best = i$                            // update DT-TIs champion
        $team_k = best$                               // add DT-TIs champion to the team
    while $Retrain == False$ do
        trade
        $t{+}{+}$
```

The following experiments were performed within this section:

- FXGP — original Base FXGP version of the algorithm as described in Section 3.1 i.e., wider range of TI and DT.

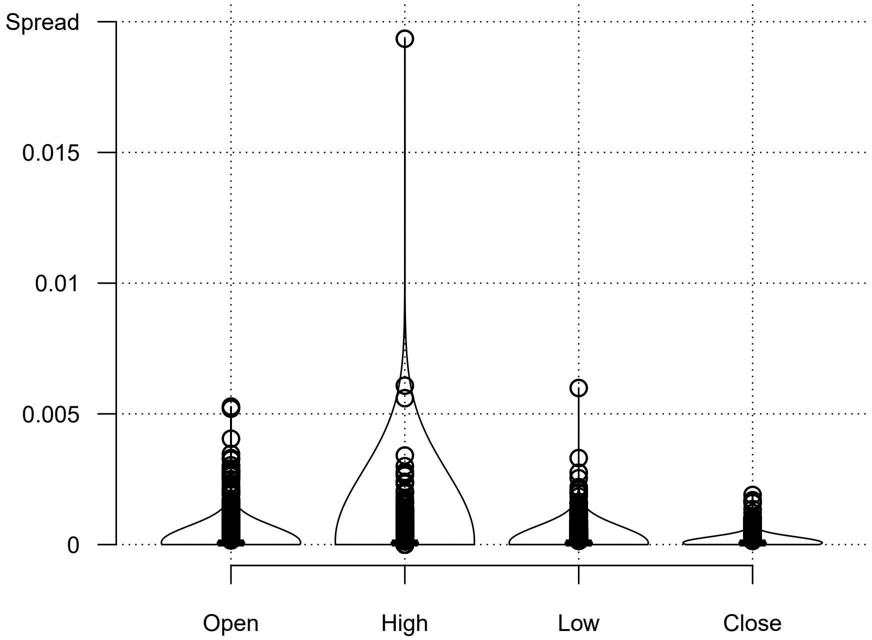


Figure 3.9: Spreads distributions. Internal box-plot provides quartile statistics. Violin profile characterizes the distribution. All spreads (0. . . 0.02 USD).

- sFXGP — FXGP *simplified/revised* version of the algorithm i.e., limited TI and DT (Section 3.3).
- FXGPT(3) — teams formed using three sFXGP champions under teaming mode 0 (Table 3.10).
- FXGPT(3e) — teams formed using three sFXGP champions under teaming mode 1 (Table 3.10)

Both updated versions of the algorithm (sFXGP and FXGPT) inherit the parameterization of the original FXGP (as described in 3.1.6) with the addition of the parameters, specific for the evolution of teams (Table 3.10). All runs were performed on a 2.8 GHz iMac computer with Intel Core i7 CPU, 16GB RAM and Mac OS X 10.7.2. Where indicated, use is also made of the Apple GCD enqueue application which identifies tasks for simultaneous execution against the available CPU cores.

Each experiment includes 100 simulation runs.

Table 3.11 provides the overview of both the number of profitable runs and the respective quartile statistics. Comparing FXGP to FXGP†indicates that removing

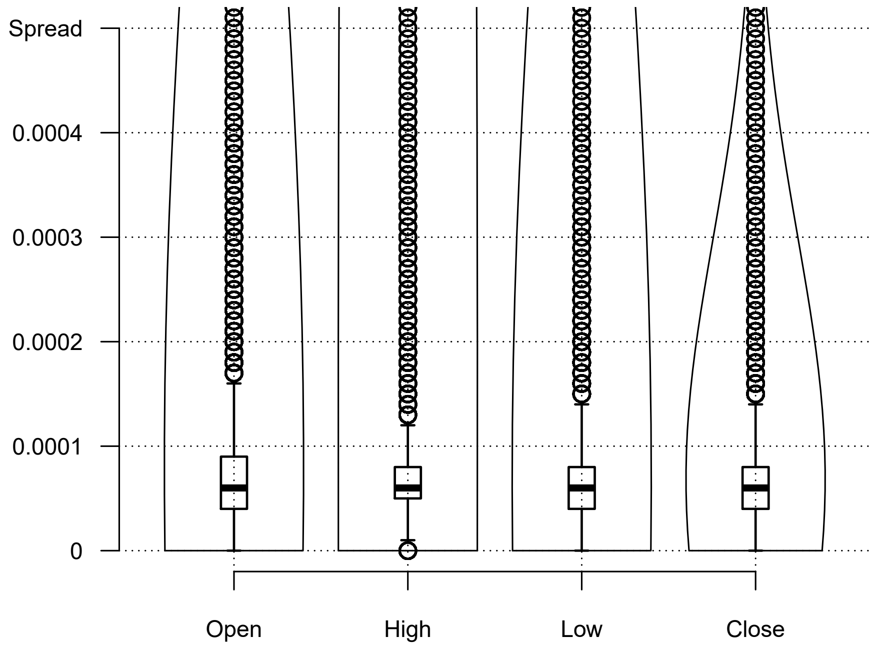


Figure 3.10: Spreads distributions. Internal box-plot provides quartile statistics. Violin profile characterizes the distribution. Spreads within the range 0. . . 0.0005 USD.

Algorithm	Profitable runs, %	Scores, pips (quartiles) min	1st	median	3rd	max
FXGP	73	-3459	-63	996	1905	4160
*s*FXGP	74	-3154	-95	989	1918	4055
FXGPT(3)	78	-2219	70	1117	2145	6291
FXGPT(3e)	81	-1877	289	1489	2463	4362
FXGP†	82	-3088	383	1523	2640	5299

Table 3.11: Quartile performance of trading agents (pips). FXGP†was the best previous result using the original FXGP algorithm with prior knowledge of spread (fixed spread of 0.00002 pips). FXGP is the same algorithm without accurate spread information (floating spreads). *s*FXGP, FXGPT(3) and FXGPT(3e) represent the proposed single agent and two 3 agent formulations.

the prior knowledge regarding spread limits results in an immediate significant reduction in performance. The simplifications introduced to provide *s*FXGP (from FXGP) have no measurable impact on the quality of trades. Introducing the simplest form of multi-agent behavior, FXGPT(3) (sampling a single champion from each independently evolved population), results in a ≈ 13% improvement to the median score. Introducing evolution using teaming mode 1 (FXGPT(3e)) results in a tightening of the distribution of scores, as well as providing a 50% improvement relative to the single agent case (Table 3.11). This also results in FXGPT(3e) managing to match the performance of FXGP†, where the latter makes use of prior information in selecting an optimal spread. The p-values for a Student t-test at the 95% confidence interval as applied between each pairwise test of FXGPT(3e) against FXGPT(3), sFXGP and FXGP are shown in the Table 3.12. And the $p-values$ ($\geqslant 0.05$) of the Shapiro-Wilk test of normality (Table 3.13) confirm the hypothesis of normality for all distributions.

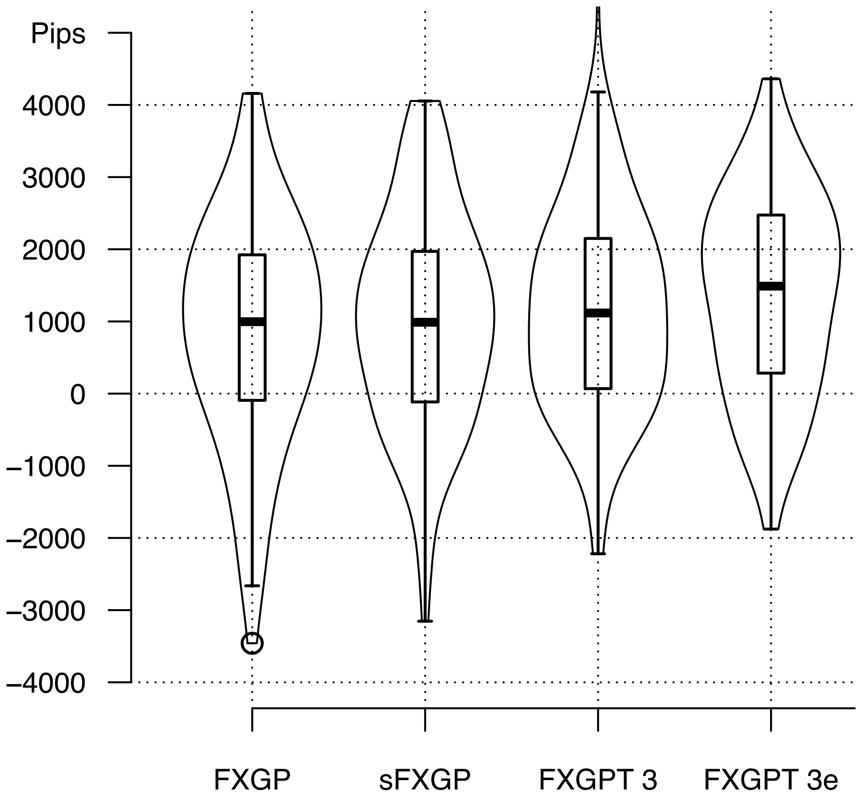


Figure 3.11: Distribution of scores in pips over trading simulation period of time. Quartile information appears in the box plot (illustrating the information from Table 3.11). The contours of the violin mimic the actual distribution of the underlying data.

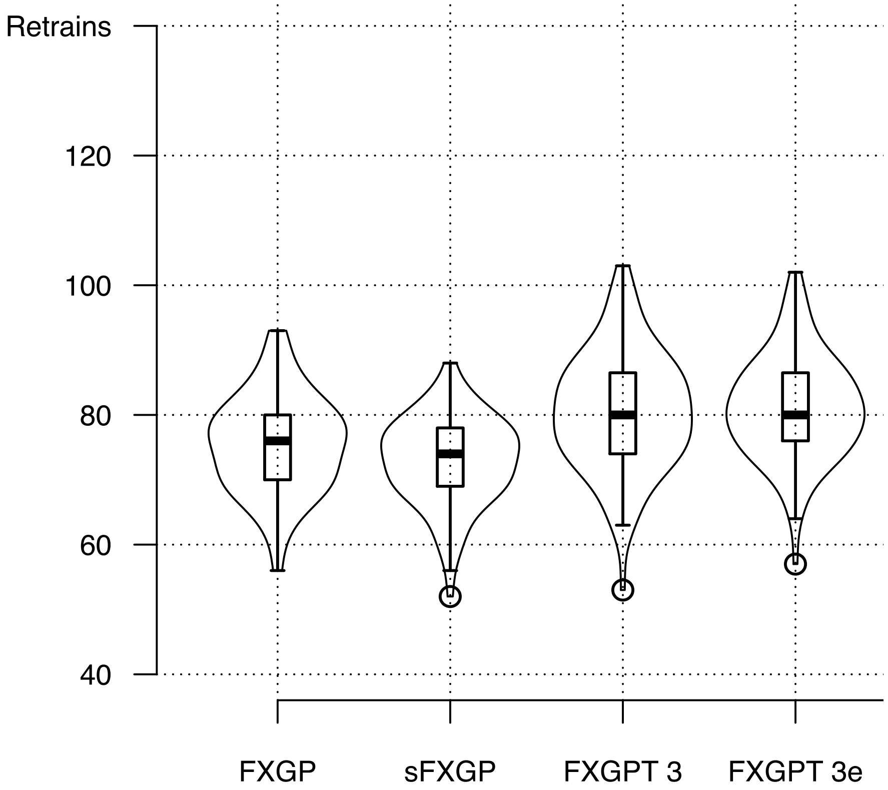


Figure 3.12: Distribution of number of retraining events over trading simulation period of time. Box plot define the quartile information and violin the actual distribution.

FXGPT(3e) vs. FXGPT(3)	FXGPT(3e) vs. *s*FXGP	FXGPT(3e) vs. FXGP
0.161	0.047	0.008

Table 3.12: *p-values* for pairwise Student t-test of FXGPT(3e) against FXGPT(3), *s*FXGP and FXGP

FXGPT(3e)	FXGPT(3)	*s*FXGP	FXGP
0.285	0.264	0.583	0.317

Table 3.13: Shapiro-Wilk test of normality, *p-values* for FXGPT(3e), FXGPT(3), *s*FXGP and FXGP cases

In order to characterize the computational costs of each algorithm, the total number of retraining events and the cost of any single retraining event is reported. Figure 3.12 summarizes the total count of retraining events over the three year trading period. In the case of both single agent algorithms (FXGP and sFXGP), a significant reduction in the number of retraining events occurs. Given that there was no trading benefit in assuming the (original) FXGP framework over sFXGP, this reduction in the number of retraining intervals appears to indicate that sFXGP agents are more general. Conversely, there is a significant increase in the number of retraining events when teams of trading agents are assumed (either mode of FXGPT).

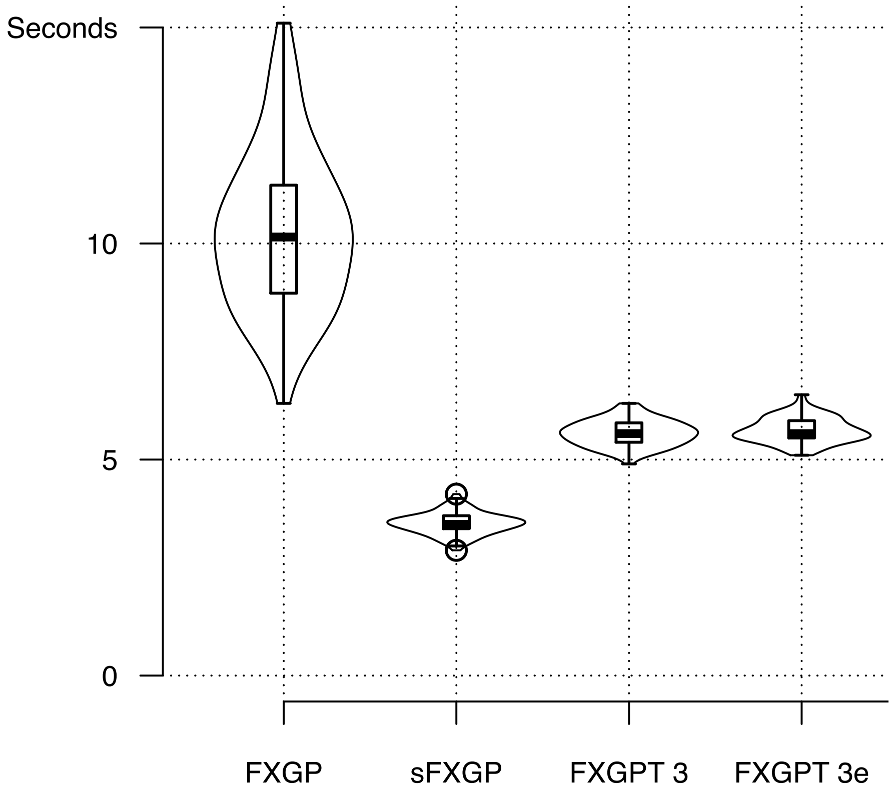


Figure 3.13: Distribution of average training times (over a run) per population (FXGP and sFXGP) and per team of three populations (FXGPT).

Figure 3.13 summarizes the cost of performing any single retraining event. It is immediately apparent that sFXGP is significantly faster than FXGP as originally conceived. Thus, the cost of supporting multiple types of moving average and division operators as well as a square root operator (TI population) does not result in any better trading performance while reducing the computational overhead by 65 – 70 % w.r.t. sFXGP. FXGPT is able to maintain the computational overhead at $\approx$ 40%, albeit with use of the coarse grained parallelism available through Apple GCD.[12]

Finally, Figures 3.14–3.18 show the statistics about the deployed sFXGP (Table 3.11) trading agents. The statistics cover 13289 deployed sFXGP agents over 100 runs. Figure 3.14 shows the distribution of returns (%) per a deployed trading agent, Figure 3.15 shows the distribution of SL and TP orders (pips) per a deployed trading agent, Figure 3.16 shows the distribution of DT sizes (nodes) per a deployed trading agent, and Figures 3.17 and 3.18 shows the distribution of number of trades per a deployed trading agent. In short, the majority of trading agents consist of 2 to 3 nodes, and participate in 4 to 13 trades, i.e. a reactive as opposed to a predictive use of technical analysis (Chapter 1). The long tail of the trades per trading agent would be more indicative of a successful predictive instance of technical analysis. That said, it is not possible to a priori identify which agents will be reactive and which would be predictive, effectively rendering all agents reactive.

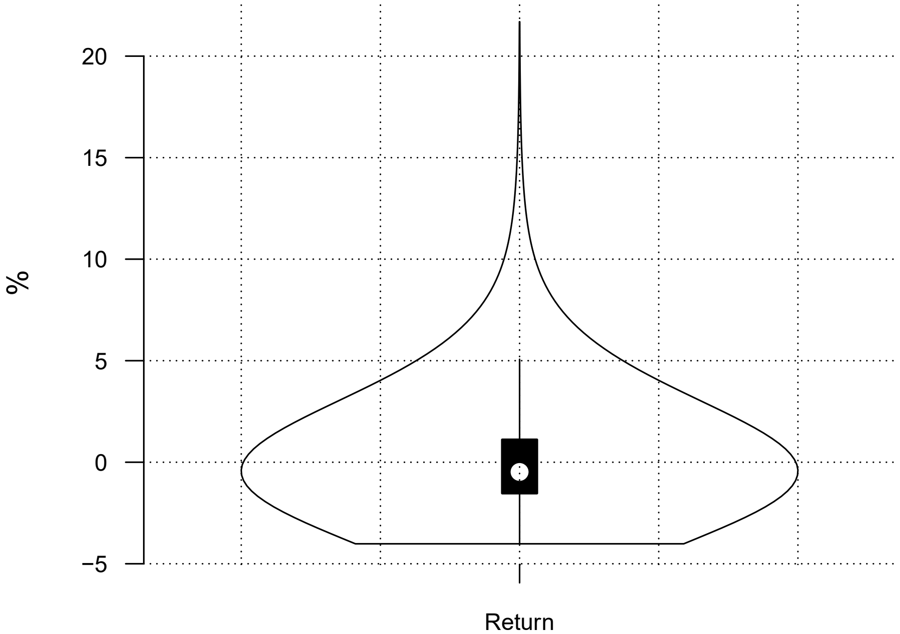


Figure 3.14: Distribution of returns (%) per a deployed trading agent (statistics collected over 100 runs) (*s*FXGP in Table 3.11). Min -4.02, 1st quartile -1.53, median -0.48, 3d quartile 1.12, max 21.69, average 0.15.

---

[12] GCD does not facilitate speeding up evaluation of a single population.

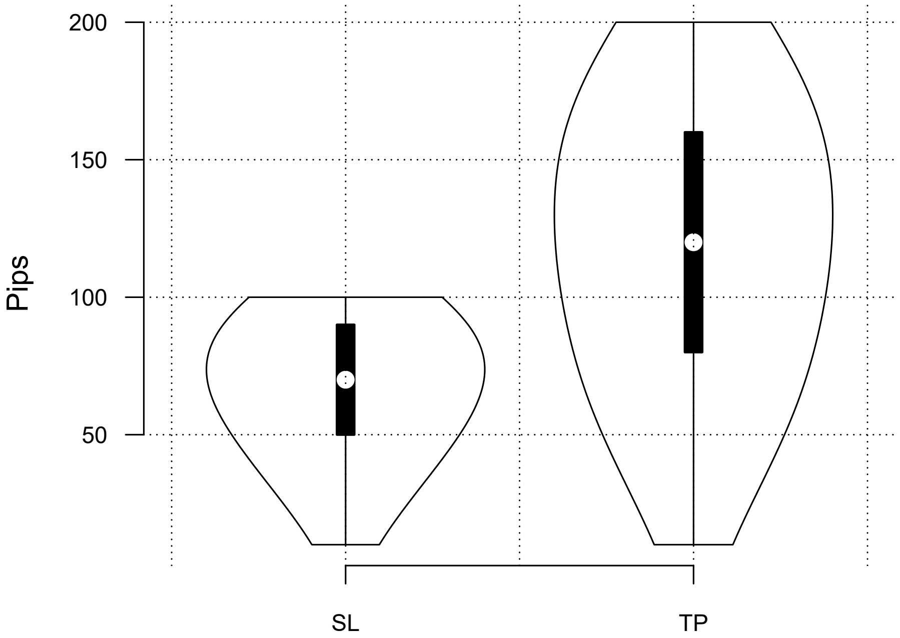


Figure 3.15: Distribution of SL and TP orders (pips) per a deployed trading agent (statistics collected over 100 runs) (sFXGP in Table 3.11). SL: min 10, 1st quartile 50, median 70, 3d quartile 90, max 100, average 66.82. TP: min 10, 1st quartile 80, median 120, 3d quartile 160, max 200, average 118.12.

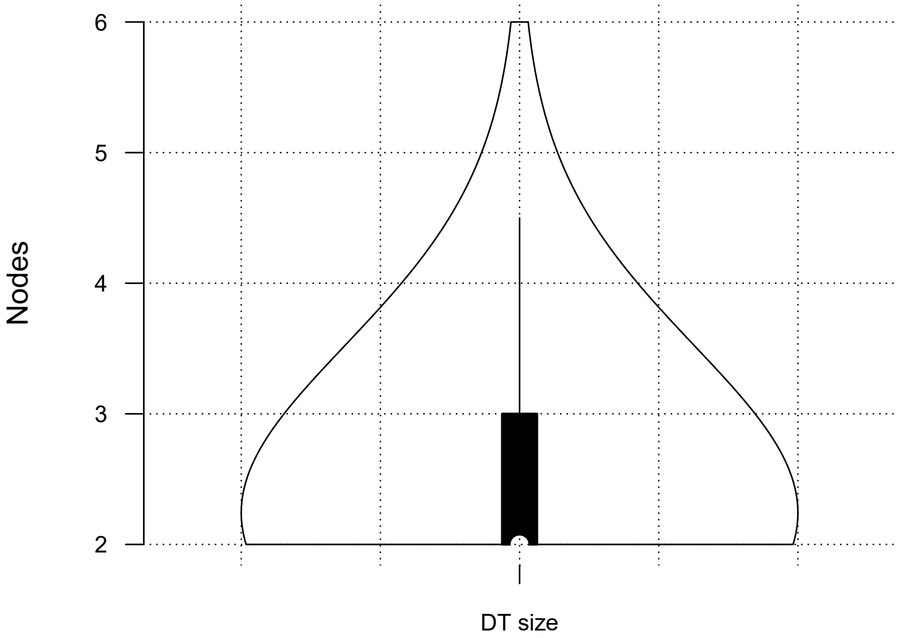


Figure 3.16: Distribution of DT sizes (nodes) per a deployed trading agent (statistics collected over 100 runs) (*s*FXGP in Table 3.11). Min 2, 1st quartile 2, median 2, 3d quartile 3, max 6, average 2.34.

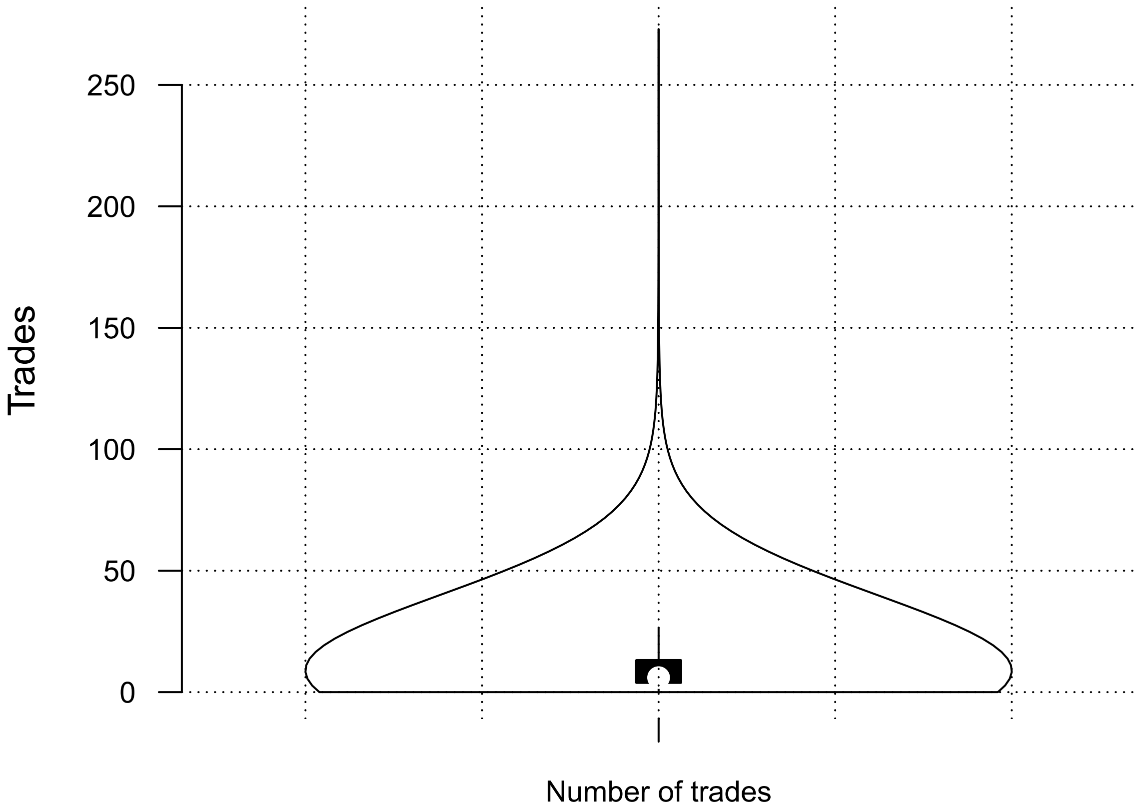


Figure 3.17: Distribution (linear scale) of number of trades per a deployed trading agent (statistics collected over 100 runs) (sFXGP in Table 3.11). Min 0, 1st quartile 4, median 6, 3d quartile 13, max 273, average 10.39.

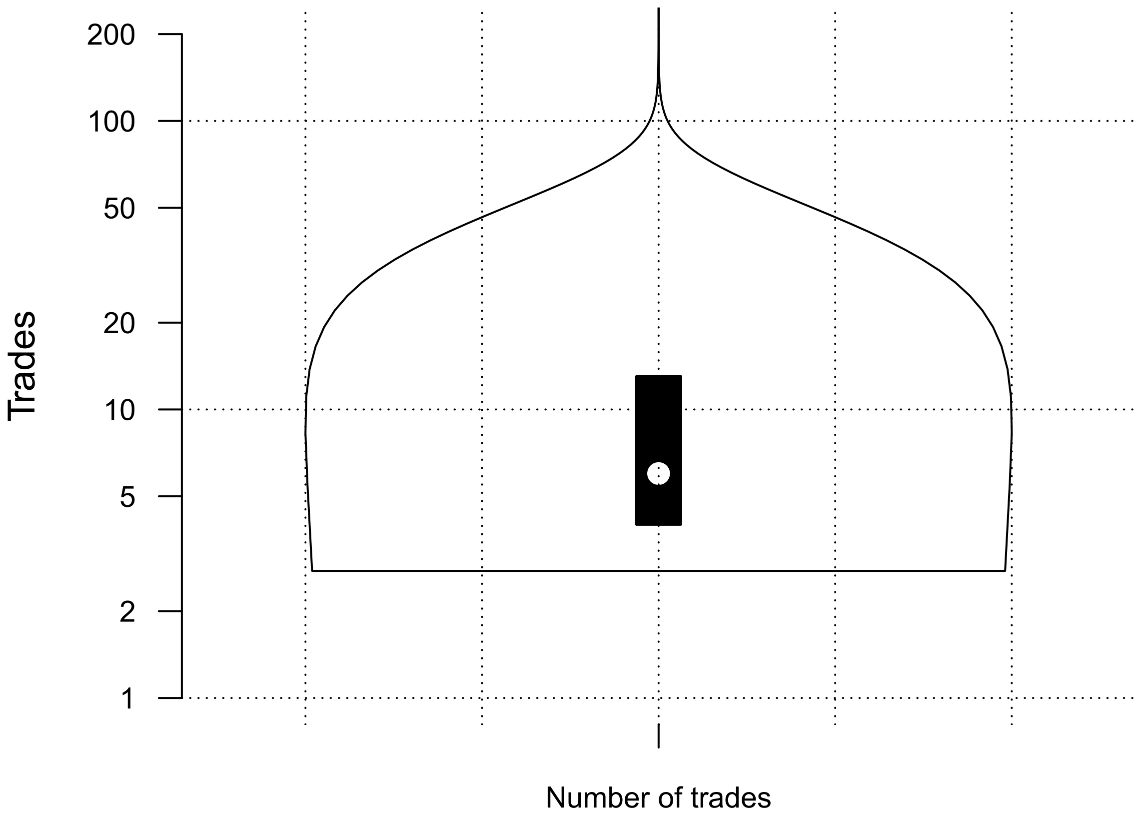


Figure 3.18: Distribution (logarithmic scale) of number of trades per a deployed trading agent (statistics collected over 100 runs) (sFXGP in Table 3.11). Min 0, 1st quartile 4, median 6, 3d quartile 13, max 273, average 10.39.

3.4 Benchmarking the Coevolution of TI and DT Versus the Evolution of DT Against a Fixed Set of TI

In this section, the FXGP algorithm will be compared with a trading system generator that utilizes a fixed set of eight popular technical indicators (fixed set of TI) [68]. The goal is, therefore, to determine the utility of employing coevolved TI.

In order to evaluate the effectiveness of the coevolution of TI and DT populations, the FXGP algorithm 3.3 was modified to obtain a benchmarking framework that evolves DT population against a fixed set of the following popular technical indicators [68]:

- SMA — Simple Moving Average.
- WMA — Weighted Moving Average.
- EMA — Exponentially Weighted Moving Average.
- RSI — Relative Strength Index.
- MACD — Moving Average Convergence Divergence.
- DMI — Directional Movement Index.
- ADX — Average Directional Movement Index[13].
- Stoch — Slow Stochastic.

The trading simulation was performed over the period of time from January 2010 to November 2012 — the same period as was used to perform the trading simulation in 3.3.2 and for the same currency pair (EURUSD, one hour time intervals). The TIs' values were calculated with popular TA-Lib library[14] that is used by many commercial trading terminals[15]. The full set of technical indicators and trading rules associated with them are listed in the Table 3.14. The three moving averages (SMA, WMA and EMA) were calculated for six periods each (three periods for the fast and three

---

[13] The ADX requires a sequence of calculations due to the multiple lines in the indicator. ADX and ADXR (Table 3.14) denote different lines of the ADX indicator implemented in the TA-Lib library.

[14] `http://ta-lib.org`

[15] `http://ta-lib.org/hdr_dev.html`

periods for the slow moving averages), all other TIs were calculated using default TA-Lib parameters (Table 3.15). Therefore, the resulting TI population includes 23 technical indicators.

TI	Trading signal (at time t)
SMA	Buy: $SMA^{fast}_{t-1} \leqslant SMA^{slow}_{t-1}$ *and* $SMA^{fast}_{t} > SMA^{slow}_{t}$ Sell: $SMA^{fast}_{t-1} \geqslant SMA^{slow}_{t-1}$ *and* $SMA^{fast}_{t} < SMA^{slow}_{t}$ Hold: *otherwise*
WMA	Buy: $WMA^{fast}_{t-1} \leqslant WMA^{slow}_{t-1}$ *and* $WMA^{fast}_{t} > WMA^{slow}_{t}$ Sell: $WMA^{fast}_{t-1} \geqslant WMA^{slow}_{t-1}$ *and* $WMA^{fast}_{t} < WMA^{slow}_{t}$ Hold: *otherwise*
EMA	Buy: $EMA^{fast}_{t-1} \leqslant EMA^{slow}_{t-1}$ *and* $EMA^{fast}_{t} > EMA^{slow}_{t}$ Sell: $EMA^{fast}_{t-1} \geqslant EMA^{slow}_{t-1}$ *and* $EMA^{fast}_{t} < EMA^{slow}_{t}$ Hold: *otherwise*
RSI	Buy: $RSI_{t-1} \leqslant 30$ *and* $30 < RSI_t < 70$ Sell: $RSI_{t-1} \geqslant 70$ *and* $30 < RSI_t < 70$ Hold: *otherwise*
MACD	Buy: $MACD_{t-1} \leqslant Signal_{t-1}$ *and* $MACD_t > Signal_t$ Sell: $MACD_{t-1} \geqslant Signal_{t-1}$ *and* $MACD_t < Signal_t$ Hold: *otherwise*
DMI	Buy: $PDI_{t-1} \leqslant MDI_{t-1}$ *and* $PDI_t > MDI_t$ Sell: $PDI_{t-1} \geqslant MDI_{t-1}$ *and* $PDI_t < MDI_t$ Hold: *otherwise*
ADX	Buy: $ADX_{t-1} \leqslant ADXR_{t-1}$ *and* $ADX_t > ADXR_t$ Sell:$ADX_{t-1} \geqslant ADXR_{t-1}$ *and* $ADX_t < ADXR_t$ Hold: *otherwise*
Stoch	Buy: $\%D < 20$ *and* $\%K_{t-1} \leqslant \%D_{t-1}$ *and* $\%K_t > \%D_t$ Sell: $\%D > 80$ *and* $\%K_{t-1} \geqslant \%D_{t-1}$ *and* $\%K_t < \%D_t$ Hold: *otherwise*

Table 3.14: Fixed set of technical indicators and associated trading rules.

3.4.1 Results of Comparison Between the Coevolution of TI and DT Versus the Evolution of DT Against a Fixed Set of TI

To compare the result of the benchmarking framework with the results of the FXGP algorithm (Section 3.3) 100 simulations were performed over the same period of time as for FXGP — from January 2010 to November 2012 using the same parameterization as in Section 3.3. The results are summarized in Table 3.16 and Figure 3.19.

TI	Parameters
SMA^{fast}	timeperiod: 4, 8 or 12
SMA^{slow}	timeperiod: 24, 48 or 72
WMA^{fast}	timeperiod: 4, 8 or 12
WMA^{slow}	timeperiod: 24, 48 or 72
EMA^{fast}	timeperiod: 4, 8 or 12
EMA^{slow}	timeperiod: 24, 48 or 72
RSI	timeperiod: 14
MACD	fastperiod: 12 slowperiod: 26 signalperiod: 9
DMI	plus_di timeperiod: 14 minus_di timeperiod: 14
ADX	adx timeperiod: 14 adxr timeperiod: 14
Stoch	fastk_period: 5 slowk_period: 3 slowk_matype: 0 slowd_period: 3 slowd_matype: 0

Table 3.15: TI parameters. All names of parameters are adopted from TA-Lib and defined in Table 3.14.

Algorithm	Profitable runs (%)	Score (pips)				
		min	1st quartile	median	3rd quartile	max
FXGP[†]	74	-3154	-95	989	1918	4055
Fixed set of TI	31	-4637	-1839	-788	386	4735

Table 3.16: Single trading agent comparison. Unpaired Student t-test *p-value = 7.066e-11*. [†] — adopted from [100].

Table 3.16 summarizes the result of conducting statistical hypothesis testing between the FXGP algorithm (coevolution of DT and TI populations) and the same framework, but limited to evolving DT alone (given a fixed set of TI). The coevolutionary framework significantly outperforms the benchmarking framework that evolves DT over a fixed set of TI (Figure 3.19). This topic will also be returned to within the context of the more challenging context of frequent intraday trading (Section 6.7).

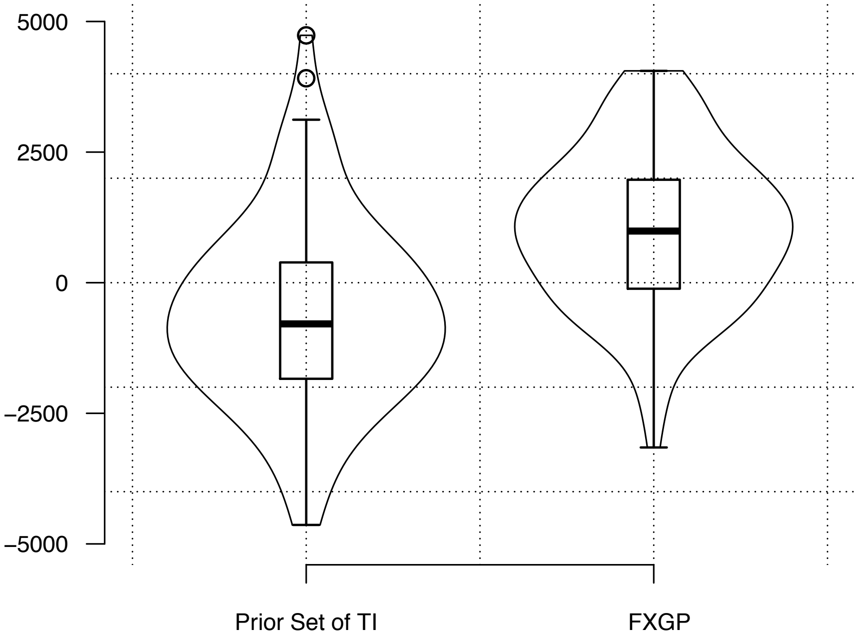


Figure 3.19: Distribution of cumulative scores in pips over trading simulation period of time.

3.5 Discussion

The following observations are made regarding the revised FXGP framework:

- The use of real prices with floating spreads (Table 3.11, *s*FXGP) significantly affects the trading results and reduces the number of profitable solutions and scores compared to trading with assumed fixed spreads as previously reported

(Table 3.11, FXGP†).

- Both single agent variants (FXGP and *s*FXGP) return a very similar number of profitable solutions and scores (Table 3.11). Moreover, the simplifications introduced to *s*FXGP did not reduce the performance of the algorithm and, at the same time, the training time was reduced by 65% (Figure 3.13). The average number of retrains was also reduced (Figure 3.12).

- The use of teams of champion trading agents (Table 3.11, FXGPT(3)) improves the negative spread of runs compared to that of a single trading agent (Table 3.11, *s*FXGP). At the same time the CPU cost for maintaining a team of champion agents is still significantly $\approx 40\%$ lower than that for the original FXGP.

- The use of evolved teams (Table 3.11, FXGPT(3e)) outperformed all other configurations and demonstrated the best results in all categories: trustability (the percentage of profitable runs) and quartile scores. Indeed, this configuration provides statistically significant improvements over the single population models (95 percentile) and adds > 350 pips to the median performance of FXGPT(3).

- Removing the capability to coevolve TI has a very negative impact on the quality of the resulting trading strategies. Thus, TI designed for the benefit of informing humans does not imply that they represent an effective starting point for an automated trading system. That said, it could be that some combination of human designed TI might be useful, it is just not possible to a priori identify such a set.

The modified multi-agent version of the FXGP algorithm (*s*FXGP and FXGPT in this section) will be referred to as FXGP hereafter.

Chapter 4

Non-Financial Streaming Data Analysis

The basic goal of this part of the research is to apply FXGP for non-financial streaming data analysis and, therefore, to identify:

1. the relevance of candlestick preprocessing to a task outside of TI for financial applications and;
2. to quantify to what degree FXGP is able to provide results comparable with those from algorithms explicitly designed for streaming data classification.

The task of predicting/forecasting the direction of movement of an indicator in a time sequence has recently received increased interest, particularly within the context of streaming data classification [54, 15, 16, 63]. It is interesting to note that although many EC approaches have been proposed for prediction within the context of financial markets (Chapter 2), there is no evidence of them also being deployed under streaming data contexts. This section utilizes the FXGP algorithm to predict the change in power consumption (increase, decrease, no change) at the next time step for the individual household electric power consumption dataset (IHEPC hereafter).

With this in mind, the prediction horizon of the FXGP algorithm was limited to the next candlestick only, unused (FXGP) functions were disabled and trading signals were converted into labels according to Table 4.1. Retraining is triggered after every ten classification errors.[1]

The original IHEPC dataset consists of household electricity consumption as measured at one-minute intervals over a period of 47 months (December 2006 — November 2010) [90].[2] This dataset has a higher resolution with respect to the units of time than the frequently employed 'Australian New South Wales Electricity Market' data

---

[1] Naturally, more formal schemes could have also been considered, such as statistical formulations from streaming data analysis [55]. However, our interest lies in the particular contributions of the candlestick preprocessing and coevolution, as opposed to the development of a change detector.

[2] `http://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption`

Trading signal	Label
Buy	‘1’
Sell	‘-1’
Hold	‘0’

Table 4.1: Trading signals to labels conversion.

(five-minute intervals) [62]. The high level of sampling available implies that multiple datasets can be constructed from the same source data by assuming different periodicities for constructing source statistics.

Table 4.3 summarizes the attributes of the original dataset, hereafter ‘Dataset 0’. The Skewness and Kurtosis metrics of ‘a1’ attribute (global active power) are shown in the Table 4.2. It is apparent that relative to the ‘baseline’ of a normal distribution, the underlying properties of the global active power dataset are a positive skew and the frequency of outliers is significantly higher. Note, however, this does not imply that consecutive time periods need be correlated, i.e. past positively skewed trends are not necessarily a guarantee of future positively-skewed trends [141]. This property will be demonstrated in the following empirical study through the use of a ‘no change classifier’ (Section 4.1.1).

Attribute	Skewness	Kurtosis
a1 (global active power)	1.78623	7.21867

Table 4.2: The Skewness and Kurtosis metrics of global active power

The ‘Dataset 0’ data is then preprocessed and converted into five separate datasets named ‘Dataset 1’ through ‘Dataset 5’ and described below. All the resulting datasets assume an ARFF format that is accepted by frameworks for data stream mining such as MOA [17]. Specifically, the goal is to predict the movement in the ‘global active power’ attribute (a3, Table 4.3).

Dataset 1 was obtained by summing the one-minute measurements within consecutive 30 minute intervals and labelling as shown in Table 4.4. Dataset 1 will represent the base case dataset.

Dataset 2 represents a characterization of the original ‘a3’ attribute using the aggregation of ‘open-high-low-close’ information over the 30 minute period (Table 4.5).

Attribute	Description (one-minute measurements)
a1	date
a2	time
a3	global active power: household global minute-averaged active power (in kilowatt)
a4	global reactive power: household global minute-averaged reactive power (in kilowatt)
a5	voltage: minute-averaged voltage (in volt)
a6	global intensity: household global minute-averaged current intensity (in ampere)
a7	sub metering 1: energy sub-metering No. 1 (in watt-hour of active energy). It corresponds to the kitchen, containing mainly a dishwasher, an oven and a microwave (hot plates are not electric but gas powered)
a8	sub metering 2: energy sub-metering No. 2 (in watt-hour of active energy). It corresponds to the laundry room, containing a washing-machine, a tumble-drier, a refrigerator and a light
a9	sub metering 3: energy sub-metering No. 3 (in watt-hour of active energy). It corresponds to an electric water-heater and an air-conditioner

Table 4.3: Dataset 0 — Original individual household electric power consumption. Attribute information.

This corresponds to the widely employed ‘candlestick’ representation for price data in financial or stock data as assumed earlier (i.e. a preprocessing step that potentially reduces the amount of noise in the original measurements). Note that the criteria for the label are still relative to the definition for attribute ‘b3’ in Dataset 1.

Dataset 3 assumes the same preprocessing of the attributes into 30 minute candlestick’s, but casts the labelling task into one of three categories (less, approximately the same, or more) as opposed to one of two. The overall distribution of class labels is summarized in Table 4.7.

Dataset 4 and Dataset 5 were processed in the same way as Dataset 2 and Dataset 3 respectively, but with candlesticks estimated over consecutive 15 minutes intervals. Table 4.7 provides a summary of the static properties of each dataset (number of attributes, instances, etc).

Attribute	Description
b1	date (first ‘a1’ value within 30 minutes interval)
b2	time ‘t’ (first ‘a2’ value within 30 minutes interval)
b3	sum of a3 within 30 minutes interval
b4	sum of a4 within 30 minutes interval
b5	sum of a5 within 30 minutes interval
b6	sum of a6 within 30 minutes interval
b7	sum of a7 within 30 minutes interval
b8	sum of a8 within 30 minutes interval
b9	sum of a9 within 30 minutes interval
label	‘1’ $IF\ b3(t+1)\ \geq\ b3(t)$ ‘-1’ otherwise

Table 4.4: ‘Dataset 1’ dataset. Attribute information (30 minutes grouping).

Attribute	Description
c1	date (first ‘a1’ value within 30 minutes interval)
c2	time ‘t’ (first ‘a2’ value within 30 minutes interval)
c3	first ‘a3’ value within 30 minutes interval
c4	highest ‘a3’ value within 30 minutes interval
c5	lowest ‘a3’ value within 30 minutes interval
c6	last ‘a3’ value within 30 minutes interval
label	‘1’ $IF\ b3(t+1)\ \geq\ b3(t)$ ‘-1’ otherwise

Table 4.5: Dataset 2 — Attribute information (30 minute candlesticks, 2 class classification). Note that attributes c3 through c6 represent those potentially indexed by a TI

Attribute	Description
c1	date (first ‘a1’ value within 30 minutes interval)
c2	time ‘t’ (first ‘a2’ value within 30 minutes interval)
c3	first ‘a3’ value within 30 minutes interval
c4	highest ‘a3’ value within 30 minutes interval
c5	lowest ‘a3’ value within 30 minutes interval
c6	last ‘a3’ value within 30 minutes interval
label	‘1’ $IF\ b3(t+1)\ >\ 1.1 * b3(t)$ ‘-1’ $IF\ b3(t+1)\ <\ 0.9 * b3(t)$ ‘0’ otherwise

Table 4.6: Dataset 3 — Attribute information (30 minutes candlesticks, 3 class classification). Note that attributes c3 through c6 represent those potentially indexed by a TI

Dataset	Attributes	Instances	Classes	Class distribution
Dataset 0	9	2075259	n/a	n/a
Dataset 1	9	68320	2	32893, 35427
Dataset 2	6	68320	2	32893, 35427
Dataset 3	6	68320	3	23946, 25181, 19193
Dataset 4	6	136632	2	65924, 70708
Dataset 5	6	136632	3	46649, 48622, 41361

Table 4.7: Datasets summary.

4.1 Results of Application of FXGP to Non-Financial Streaming Data Analysis

Each preprocessed dataset was divided in two parts. The data from the end of 2006 to December 2007 was used to define the FXGP parameterization, whereas the data from January 2008 to November 2010 was used for the data stream experiments. Thus, the first DT–TI champion was deployed, starting from January 1, 2008, whereas the last week of the 2007 was used to train and validate agents during the very first Train–Validate–Label cycle (336 and 672 candlesticks for 30-minute and 15-minute candlesticks respectively) in case of FXGP and to do initial training of MOA classifiers. The performance is measured with streamAUC metric (AUC hereafter) that was introduced in [43] (see also [101]) and over duration of data stream has the form:

$$streamAUC = \frac{1}{T} \sum_{t=[1,\ldots,T]} DR(t) \tag{4.1}$$

where $DR(t)$ is a multi-class detection rate (DR) at time t and is calculated as:

$$DR(t) = \frac{1}{C} \sum_{c=[1,\ldots,C]} DR_c(t) \tag{4.2}$$

where $DR_c(t)$ is a per-class detection rate at time t and is calculated as:

$$DR_c(t) = \frac{tp_c(t)}{tp_c(t) + fn_c(t)} \tag{4.3}$$

where t is the exemplar index, and $tp_c(t)$, $fn_c(t)$ are the respective online counts for true positive and false negative rates for class 'c' up to this point in the stream.

Note that streamAUC represents an arithmetic process for approximating the 'area under the curve' and is different from the receiver operating characteristic (ROC) [72]

The parameterization with the default values that were used during the experiments is shown in Table 4.8.

4.1.1 Impact of Candlestick Preprocessing

The FXGP algorithm assumes data in the form of price time series, i.e. data preprocessed as candlesticks (datasets Data 2. . . Data 5, Table 4.7). Several models from

Parameter	Description	Default
TIp	Minimum TI population size	100
TIs	Maximum TI program size, steps	6
Regs	Number of TI program registers	2
DTp	DT population size	100
DTgap	Number of DTs replaced in each generation	25
DTmut	Relative probability of DT or TI mutation	0.5
DTs	Maximum DT size, nodes	6
Gmax	Maximum number of generations	1000
Nt	Training partition size	672. . . 13334
Nv	Validation partition size	336. . . 6666
τ	Training plateau length, generations	200
hitMax	Maximum number of errors to retrain	10

Table 4.8: FXGP parameterization.

the open-source MOA framework [17] are used to characterize the effectiveness (or otherwise) of the data pre-processing into candlesticks. Specifically, the 'No Change' classifier, Naive Bayes and Hoeffding Trees will be used.

The **No Change classifier** represents a 1-bit state-machine in which the 'prediction' reflects that of the last prediction as long as the last prediction was correct. A missed prediction results in the state changing to predict the new class. Such a predictor does not make any use of the attribute information, only knowledge of the labels. Previous research has demonstrated that such a naive model is capable of surprisingly strong performance when there is a low amount of mixing (turnover) in consecutive labels [16].

Both the Naive Bayes and Hoeffding Tree classifiers represent well known algorithms for streaming data classification and appear in a number of monographs [54, 15]. In particular, the **Naive Bayes** model makes use of change detection to formulate when to concentrate updates to the model [55], whereas the **Hoeffding Tree** makes use of statistical sequence analysis to characterize under what conditions the decision tree is developed. Specifically, the number of observations necessary to provide an a priori level of predictive accuracy associated with an attribute is identified mathematically through the Hoeffding bound [45]. As such, this gives the Hoeffding Tree the ability to construct temporal features, a property that FXGP also addresses through the use of the TI population.

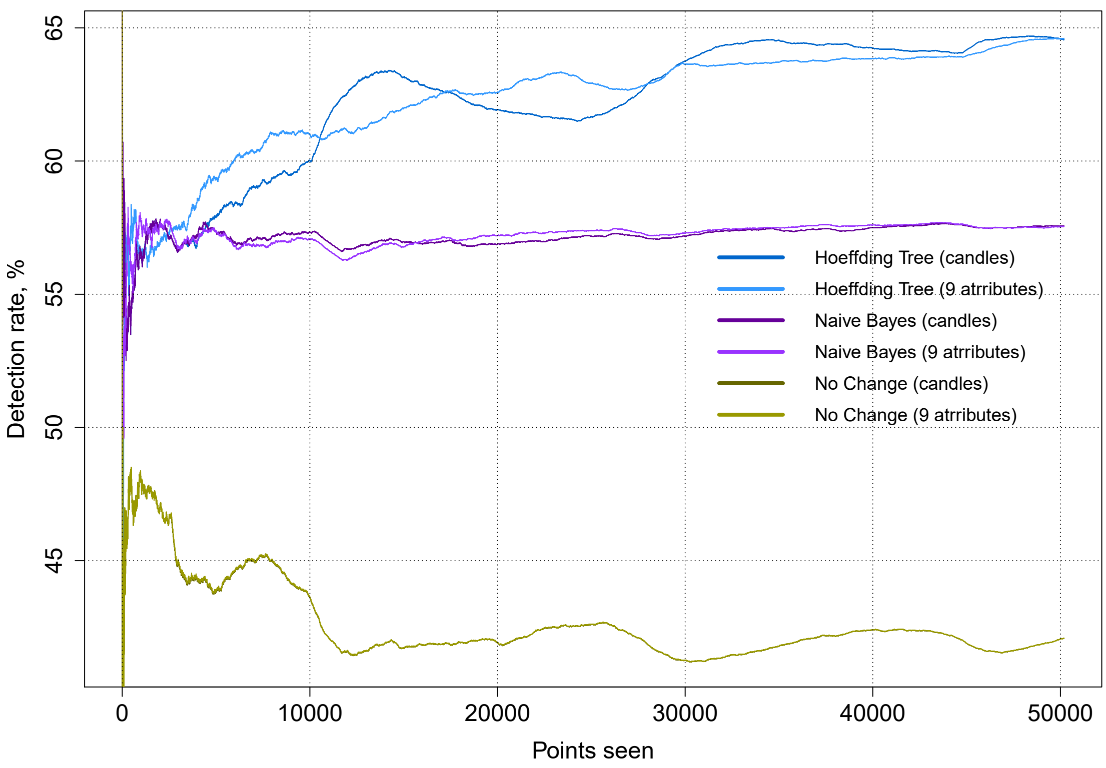


Figure 4.1: MOA framework. Full set of attributes (Dataset 1) vs 30 minutes candlesticks (Dataset 2).

The results obtained in the case of all available attributes (Dataset 1) and 30-minute candlesticks are shown in Figure 4.1. Naturally, the No Change classifier is not impacted by the preprocessing of the attribute data (it never employs any attribute information). Conversely, the Naive Bayes classifier and Hoeffding Tree do not show any particular preference for or against candlestick preprocessing. Expressing this using the stream AUC metric from Equation 4.1 indicates that this does not affect the effectiveness of all three classifiers (Table 4.9). Hereafter the candlestick preprocessing for the remainder of the study (Dataset 2 through 5) will be assumed.

4.1.2 Dataset 2 and 3 — 30 Minute Candlesticks

Each experiment includes a single run for MOA (‘Hoeffding Tree’, ‘Naive Bayes’ and ‘No Change’) and 100 independent runs for FXGP. The results for binary classification of 30-minute candlesticks are shown in Figure 4.2 and the results of ternary

Classifier	Dataset	AUC
MOA Hoeffding Tree	Dataset 2	0.623
MOA Hoeffding Tree	Dataset 1	0.623
MOA Naive Bayes	Dataset 2	0.572
MOA Naive Bayes	Dataset 1	0.572
MOA No Change	Dataset 2	0.426
MOA No Change	Dataset 1	0.426

Table 4.9: AUC summary statistics. Raw data versus Candle preprocessing

Classifier	binary (Dataset 2)	ternary (Dataset 3)
FXGP best	0.684	0.554
FXGP median	0.673	0.540
FXGP worst	0.667	0.532
MOA Hoeffding Tree	0.623	0.493
MOA Naive Bayes	0.572	0.459
MOA No Change	0.426	0.354

Table 4.10: AUC summary statistics, Dataset 2 and 3: 30 minute candlesticks, binary and ternary classification

classification (30-minute candlesticks) are shown in Figure 4.3. Table 4.10 details the streaming AUC statistic for both binary and ternary classification tasks.

Given the formulation adopted for labelling the data, adding a third class will only increase the potential for label mixing relative to the binary case, hence the reduction in performance as measured by the stream AUC statistic reflects this bias. Indeed, all classifiers return a reduction in detection rate when going from the binary to ternary formulation. The relative ranking of the models (in terms of Detection Rate) between each formulation of the dataset remains unchanged; in particular, No change < Naive Bayes < Hoeffding Tree < FXGP.

4.1.3 Dataset 4 and 5 — 15 Minute Candlesticks

The results of assuming preprocessing using the 15-minute candlesticks are shown in Figure 4.4 and Figure 4.5, for the binary and ternary classification tasks respectively. Table 4.11 summarizes the resulting quantification as reflected by the stream AUC metric.

The No Change class classifier again represents the worst case detection rate

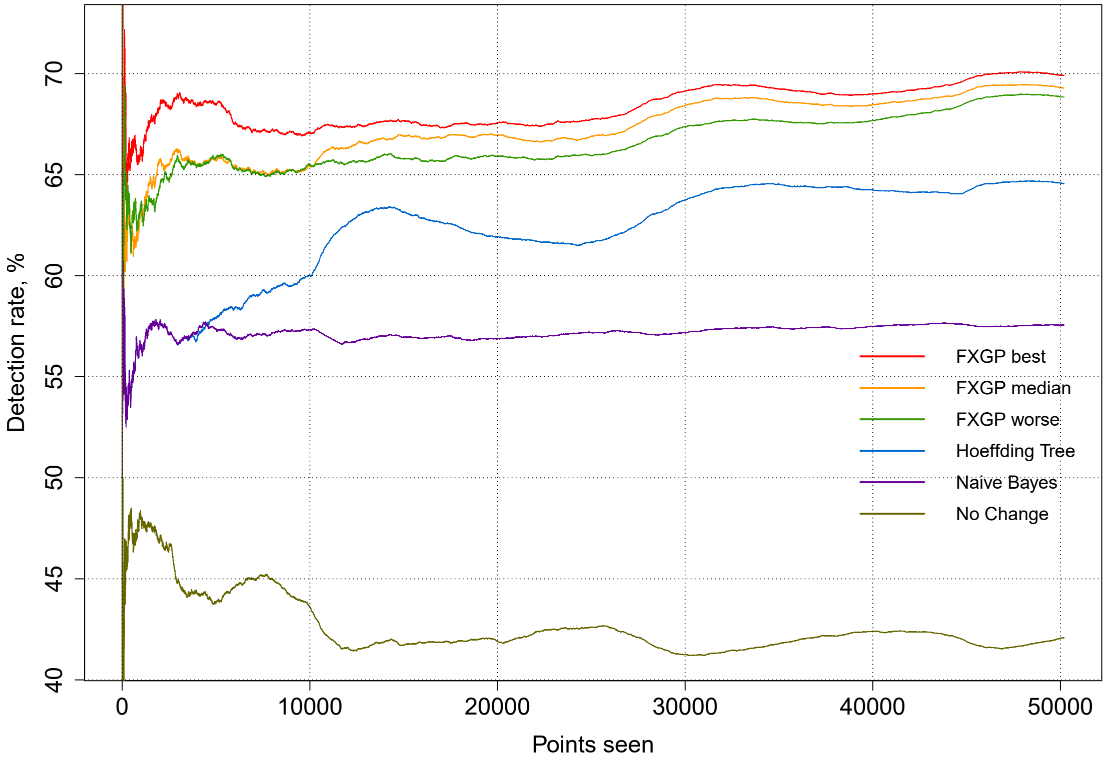


Figure 4.2: Dataset 2: 30 minute candlesticks, binary classification task.

Classifier	binary (Dataset 4)	ternary (Dataset 5)
FXGP best	0.685	0.566
FXGP median	0.675	0.559
FXGP worst	0.675	0.550
MOA Hoeffding Tree	0.625	0.503
MOA Naive Bayes	0.563	0.468
MOA No Change	0.443	0.358

Table 4.11: AUC summary statistics, Dataset 4 and 5: 15 minute candlesticks, binary and ternary classification

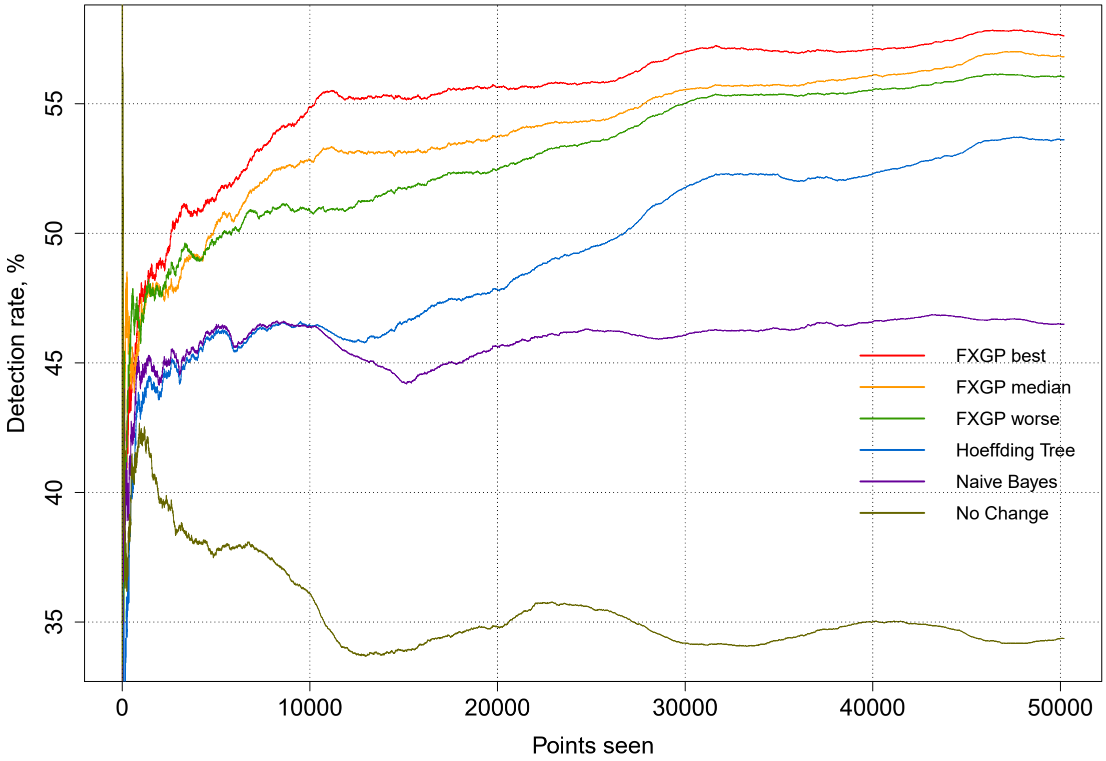


Figure 4.3: Dataset 3: 30 minute candlesticks, ternary classification task.

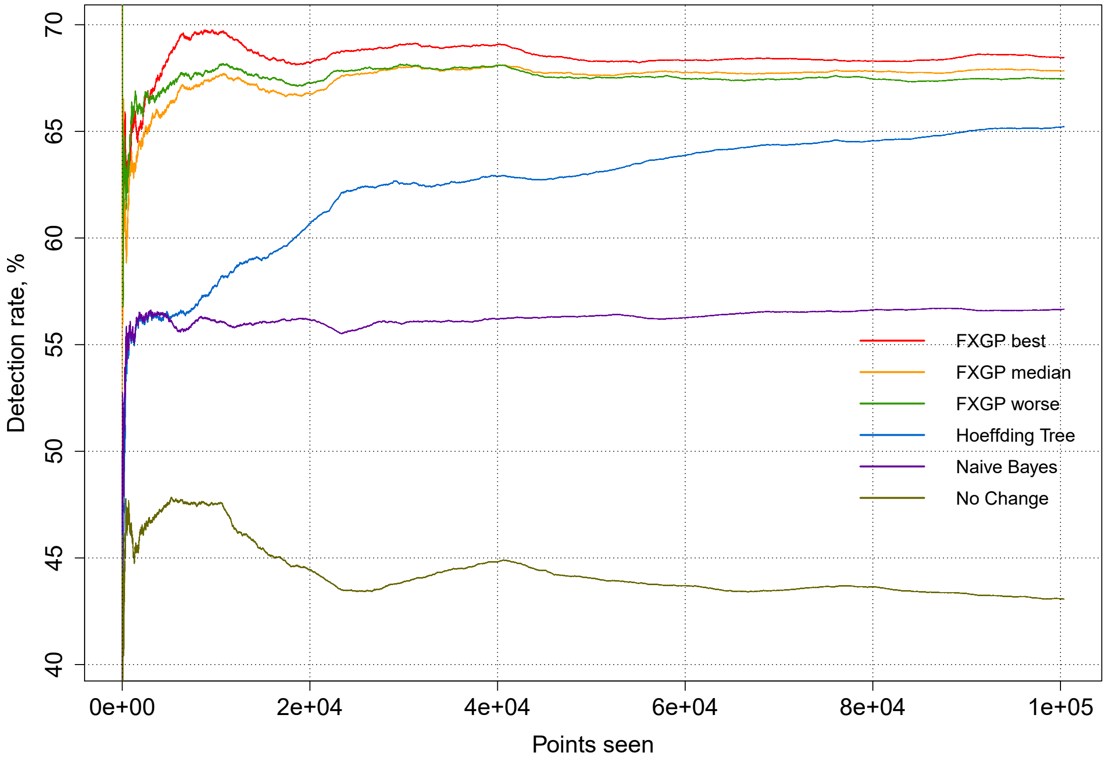


Figure 4.4: Dataset 4: 15 minute candlesticks, binary classification task.

throughout. Likewise, the relative ranking of the other models is unchanged from the case of candlesticks estimated over a 30 minute interval. However, what is now interesting is that comparing FXGP performance using 15 minute candlesticks and 30 minute candlesticks returns an increase in performance when estimating the candlestick over the shorter period. Indeed, testing for the significance of this using an unpaired two-tailed student t-test (99% confidence interval) indicates that a statistically significant improvement appears in the case of FXGP on Dataset 4 versus 2 and Dataset 5 versus 3.[3] The *p-values* ($\geqslant 0.05$) of the Shapiro-Wilk test of normality (Table 4.12) confirm the normal distribution of results in case of all datasets (Datasets 2. . . 5).

Also of note is that the eventual DR at the end of the stream might be higher in the case of the 30-minute scenarios under FXGP (compare Figures 4.2 to 4.4 and likewise Figures 4.3 to 4.5). However, in the case of the 15-minute scenarios the point

---

[3] Corresponds to a p-value $< 2.2 \times 10^{-16}$ in both cases.

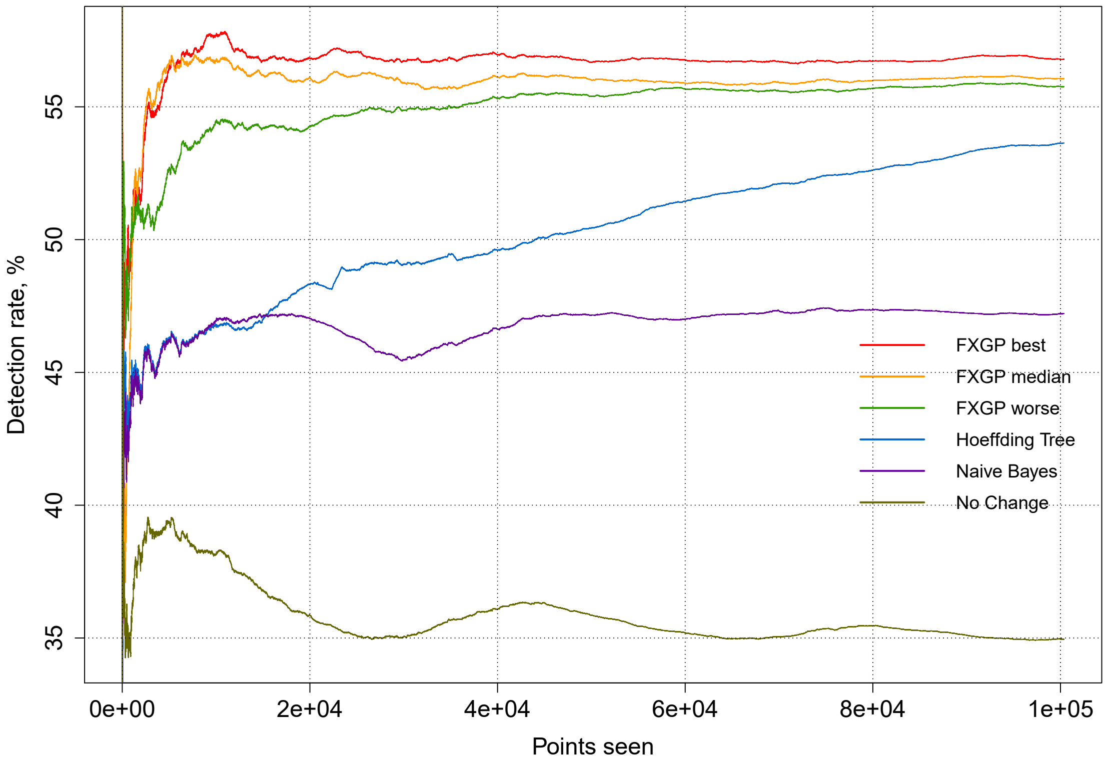


Figure 4.5: Dataset 5: 15 minute candlesticks, ternary classification task.

Dataset 2	Dataset 3	Dataset 4	Dataset 5
0.5504	0.4821	0.5638	0.7879

Table 4.12: Shapiro-Wilk test of normality, *p-values* of classification results of four cases (Datasets 2. . . 5)

in the stream at which, say, a 55% DR is first reached is much earlier. Given that there is little or no regression in DR, the corresponding stream AUC is significantly higher.

4.1.4 Quantifying the Role of Retraining

Sections 4.1.1 to 4.1.3 implicitly assumed that the individual household electric power consumption dataset would benefit from retraining or an explicitly online/ streaming approach to model building. In order to provide some quantification for this perceived benefit the retraining step for FXGP is explicitly turned off . Thus, the first 'Train—Validate' cycle is performed (relative to the same one week of data) and thereafter the DT–TI champion identified during the Validate stage is deployed to make the predictions thereafter. Figure 4.6 reflects the distribution of DR across the remainder of the stream (30 minute candlestick, binary classification).

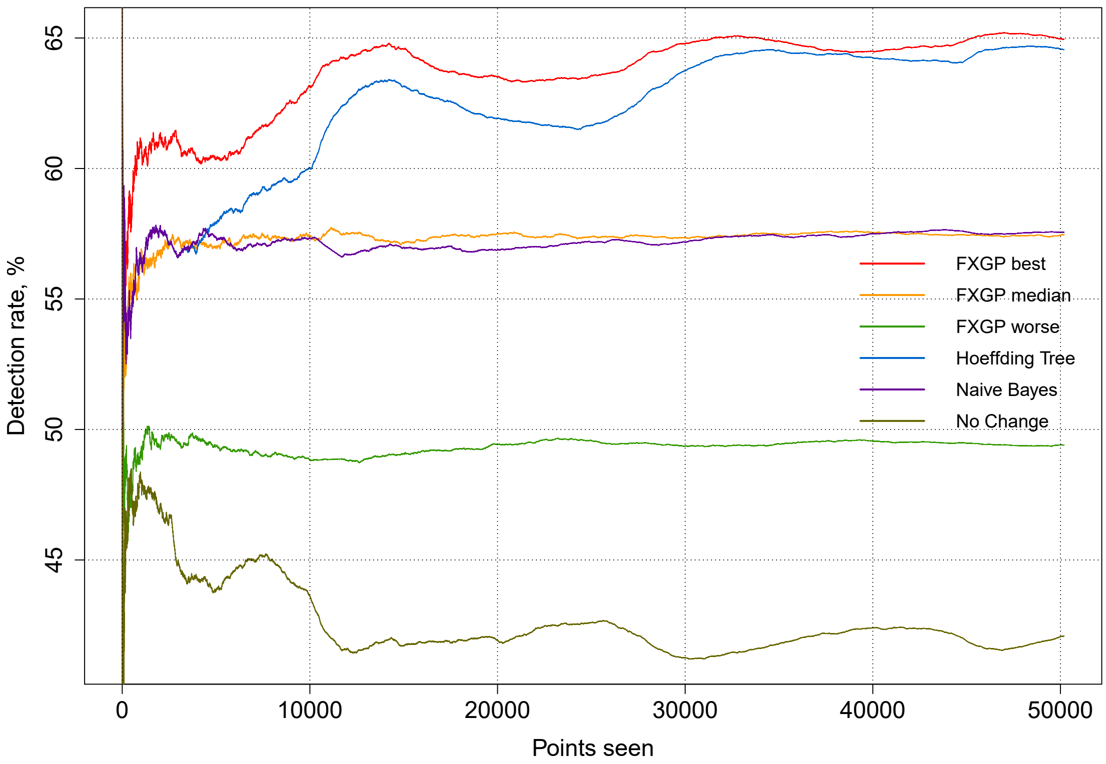


Figure 4.6: Dataset 2: 30 minute candlesticks, binary classification task — 100 runs of FXGP *without* retraining. Hoeffding Tree, Naive Bayes and No Change classifier retain retraining.

Note that for comparative purposes, DR for the curves for Hoeffding Tree, Naive Bayes and No Change classifier still reflect training *throughout* the stream, whereas the FXGP curves reflect performance without retraining. Previously, FXGP was able to return worst case performance that exceed the best baseline model. Now, without retraining, best case performance fails to reach that of the Hoeffding Tree and worst case performance is considerably worse than Naive Bayes (although still better than the No Change baseline). Indeed, the worst case profile reflects a detection rate of $\approx$ 50% or no better than labelling the data all one class. The wide variation in performance of FXGP reflects the difficulty in choosing a model when it is not possible to guarantee that the training data is representative of the underlying task (as is the case when non-stationary properties exist). Moreover, note that this is the same set of GP models that when retraining is enabled perform better than all baseline streaming classifiers.

4.2 Quantifying the CPU Cost of Retraining

FXGP does not train incrementally, but when retraining is triggered, then the content of TI and DT populations are both reset and evolution begins from a completely new initialization of random individuals. Such an approach was assumed following a benchmarking study comparing incremental evolution (some or all of the population is retained between evolutionary cycles) to the 'flush-and-restart' methodology assumed here [97]. Naturally, this also has implications for the (computational) cost of rebuilding a champion solution, potentially setting a limit to the degree to which real-time operation can be supported. Figure 4.7 quantifies this cost from the perspective of the CPU time to coevolve an entirely new DT–TI pair .[4]

Given that the time between retraining events is several orders of magnitude lower than the interval between new data samples (15 or 30 minutes) it is readily apparent that FXGP is capable of real-time operation under this task domain. Naturally, the number of re-trigger events is a function of the number of classes (difficulty of the task) and cardinality of the data stream, but in all cases remains $< 1\%$ of stream content. Thus, between ≈ 200 to ≈ 1000 retraining events are sufficient to maintain synchronization with the non-stationary properties of the stream. However, without retraining, it is generally not possible to identify good predictors (Figure 4.6).

---

[4] 2.8 GHz iMac, Intel Core i7 CPU.

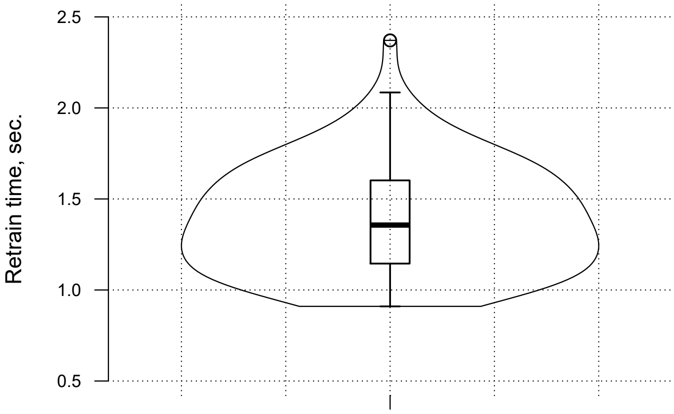


Figure 4.7: CPU time in seconds necessary to coevolve a completely new DT–TI champion. Distribution estimated over 100 runs.

4.3 Discussion

Based on the results the following conclusions can be made:

- FXGP with retraining typically out performs prequential (predictive sequential) classifiers based on the widely used Hoeffding Tree and Naive Bayes formulations. Specifically, the Hoeffding Tree framework decomposes the classification task into two processes. A statistical characterization identifies when sufficient data is available to revise the classifier and the classifier is represented as a decision tree. Conversely, FXGP uses the TI population to construct temporal features, and again defines classification in terms of a decision tree.

- GP can be applied to streaming data classification tasks and remain computationally feasible without recourse to specialist hardware/software support (e.g. no use was made of multi-threading or multi-core execution).

- Preprocessing data using a candlestick representation did not improve on the original form. However, this might also imply that the data is less noisy than experienced in financial setting. Naturally, the candlestick representation assumes that there is sufficient data present in the stream for construction of each candlestick.

Chapter 5

SL and TP Orders Verification With Fibonacci Levels (Specialized Functionality)

The main goal of work presented in this chapter is to improve the results of the proposed FXGP algorithm by proposing and assessing an approach for introducing limit orders (SL and TP) with the support and resistant levels often used by traders. This represents a generalization of **issue 6** identified in Section 3.2). The level that lies in the direction of the price trend is known as the 'resistance' level, and the level that lies in the direction opposite to the price trend is known as 'support.' The identification of support and resistance levels characterize the direction of movement of a price and therefore form the basis for predictive sell or buy strategies. The support and resistance levels are used for the purpose of creating stop-loss and take-profit orders. For example, in case of an upward trend, support levels define a price level at which the price will likely bounce off in the case of the backwards movement and will continue an upward trend. And vice versa, the resistant levels define a price level at which the price will likely bounce back, and the upward trend will turn into a downward one. However, should the price manage to break through the support or resistance level, then it will likely to continue the movement in this direction until new support or resistance level appears. The challenge of achieving the greatest profits using these methods is to find support and resistance levels before they explicitly occur i.e., proactively determine support and resistance. A range of schemes have been proposed for this purpose, including comparisons of price against pivot points (e.g., [132]), moving average envelopes (e.g., [88, 139]), or Bollinger bands (e.g., [88, 23, 24, 33]).

This section investigates the case of introducing retracement levels to dynamically characterize the size of stop-loss (SL) and take-profit (TP) orders. Stop-Loss orders represent an *a priori* rule structured to stem further losses [104]. The converse, take-profit orders, act in the predicted direction and result in closing a trading order at a profit. Previous instances of FXGP assumed that the SL (Sections 3.1.2 and 3.3) and TP (Section 3.3) orders were evolved based on the training partition of data [98, 97]. These assumptions were continued in the extended versions of FXGP that was present in Section 3.3. FXGP successfully evolves trading strategies that both minimize the number of times that a SL order is triggered and minimize the average size of SL orders. To accomplish these goals, the SL orders were combined with the trading rules evolved by FXGP such that they are more proactive and less sensitive to specific thresholds than was previously the case. To do so, the three major schemes from which different families of SL and TP orders are derived: pivot points, moving averages, and Fibonacci ratios are considered. The proposed extension of FXGP is detailed below.

For the purpose of this research three commonly used technical indicators are selected for designing SL and TP orders: Fibonacci ratios, Pivot point and Moving average. Moving averages form the basis for the indicator known as 'Bollinger Bands,' as discussed below. The following section summarizes the basic scheme assumed for each.

5.1 Moving Average

The Moving average (MA) over a period of n candlesticks from time index i is calculated as (3.1), where C_j is a Close price (Figure 5.1).

$$MA_i = \frac{\sum_{j=0}^{n-1} C_{i+j}}{n} \tag{5.1}$$

SL and TP orders can then be defined relative to the MA, in particular MA$\pm\theta$ for some value θ to define a moving average envelope around the MA. The special case of θ set to some multiple k of the n-period standard deviation above and below the n-period MA of that series of prices gives rise to Bollinger bands. In such a case, the higher and lower band vary as a function of market volatility. In the context of genetic programming a stop-loss order might take the form of MA-θ_l for a lower bound θ_l whereas a take-profit order might take the form of MA+θ_p for a chosen upper bound of θ_p, where θ_l and θ_p are evolved thresholds. Leung *et al.* noted that the more general case of MA was preferable to Bollinger bands [88], whereas Butler and Kazakov recommended adapting the θ and n parameters on a continuous basis [23, 24].

Classically, the MA based form for a stop-loss order might take the form of MA-θ_l whereas a take-profit order might take the form of MA+θ_p (buy order or long position) or otherwise in case of sell order or short position:

- Buy order: IF $price(t) > MA(t)$ THEN $SL = MA - \theta_l$ ELSE (DO NOT TRADE)
- Sell order: IF $price(t) < MA(t)$ THEN $SL = MA + \theta_p$ ELSE (DO NOT TRADE)

where θ_l and θ_p are evolved thresholds.

Figure 5.1: Moving average example, where the red line indicates MA relative to the candlestick price statistic. Image produced using the MetaTrader 4 Forex trading platform `https://www.metatrader4.com`

5.2 Fibonacci Ratios

Fibonacci ratios used as a trading strategy that involves determining support and resistance levels based on the Fibonacci sequence. In particular, the ratios are derived by dividing a number in the Fibonacci sequence by some following number in the sequence. These ratios are then used as the divisor for the distance between two extreme points on the chart. Fibonacci ratios are frequently observed to be used with various trading strategies for identifying trends' turning points and SL and

TP order prices. Fibonacci ratios should not be taken to be exact indicators of support/resistance levels. They appear to approximately coincide with points of retracement, but need to be tuned when reading the market given a trader's risk and investment goals [139]. In general, any system that attempts to construct rules of investment based on support and resistance levels need to be capable of adapting and revising the rules as the market conditions will undoubtedly change over time. Many applications for currency trading therefore include Fibonacci-derived levels for retracement, but leave the interpretation of where to make 'stop-loss'/'take-profit' decisions to the trader.

Fibonacci ratios (Fibo) may be used to define the position against which a SL or TP order is expressed. Fibonacci ratios or levels for trading (Figure 5.2) most often involve the use of the following typical cases [139]:

1. Key levels: 0 (0%), 0.236 (23.6%), 0.382 (38.2%), 0.618 (61.8%) and 1 (100%).
2. Other levels: 0.5 (50%) is derived by dividing the number 1 (third number in the Fibonacci sequence) by 2 (forth number in the Fibonacci sequence), 0.764 (76.4%) is derived from the levels 1 and 0.236 as follows: $0.764 = 1 - 0.236$
3. Extension levels: 1.618 (161.8%), -0.618 (-61.8%)

The 0 and 100 levels are identified through recent historical low and high prices (Figure 5.2). New low or high prices result in a recalculation of the intervening Fibonacci levels in a high-to-low or low-to-high trend. Figure 5.2 shows Fibonacci levels between 0 and 100 drawn through recent significant low and high prices. Note how prices appear to drift down/up to these levels before 'pushing' through. One can also note that a strong retracement occurs after the first instance of the Fibonacci level of 100. A red line joins the lowest to the maximum price point that are used to establish the Fibonacci levels.

Figure 5.2: Illustration of Fibonacci retracement. Image produced using the MetaTrader 4 Forex trading platform `https://www.metatrader4.com`

5.3 Pivot Point

The Pivot point (P) is the average of the *High* (H), *Low* (L) and the *Close* (C) prices of the previous trading session. This information is used to provide candidate support and resistance levels that then can be utilized while setting the SL and TP orders (Figure 5.3). Typical definitions for pivot point and corresponding support and resistance levels are defined as follows [132]:

- Pivot point: $P = (H + L + C)/3$
- Resistance Level 1 (R1): $R1 = 2 \times P - L$
- Support Level 1 (S1): $S1 = 2 \times P - H$
- Resistance Level 2 (R2): $R2 = P + H - L$
- Support Level 2 (S2): $S2 = P - H + L$
- Resistance Level 3 (R3): $R3 = P - S1 + R2$
- Support Level 3 (S3): $S3 = P - R2 + S1$

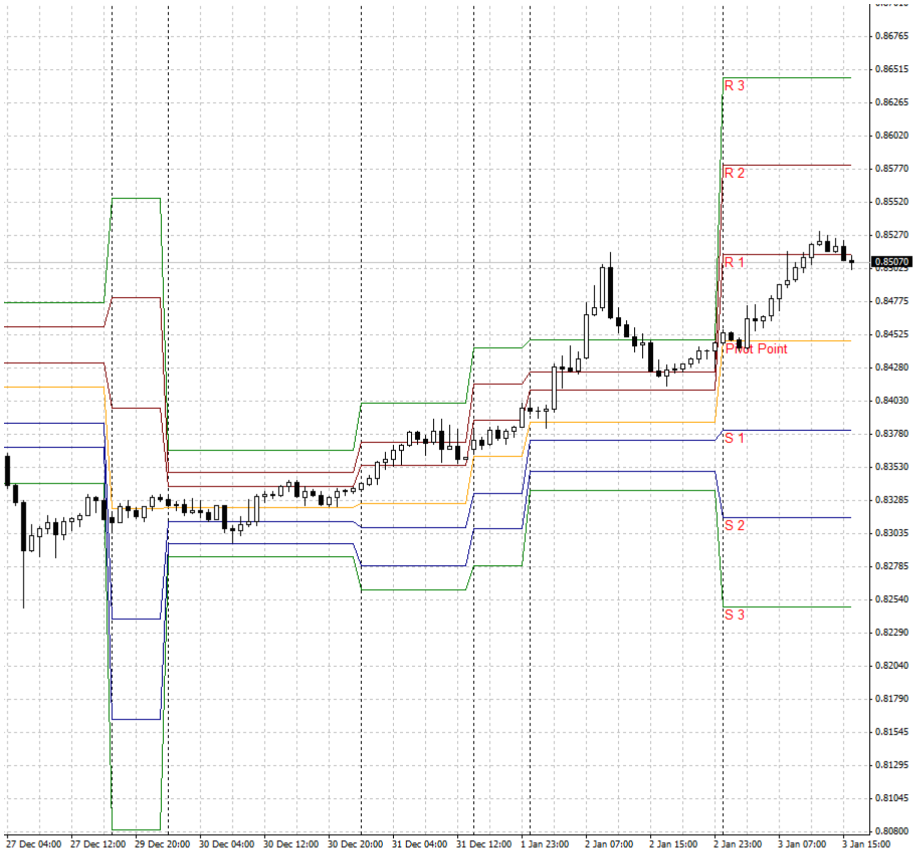


Figure 5.3: Example of support and resistance levels defined using Pivot Points. Image produced using the MetaTrader 4 Forex trading platform `https://www.metatrader4.com`

5.4 FXGP With Trading Orders Validation

Whenever a sell or buy action is generated, it is verified against one of four forms of SL or TP order, as follows:

- TP orders. When a trading agent generates a *buy* or *sell* action, FXGP can set a TP order along with the *buy* or *sell* and SL orders. The minimal size of the TP order (s_{min}) is defined by the user. Please see Table 5.1.
- Fibonacci based SL orders verification mode. In this case the SL order is verified by the Fibo levels. The *buy* and *sell* rules are described in the Table 5.2.
- Pivot-based SL orders verification mode. In this case the SL order is verified by the Pivot levels. The *buy* and *sell* rules are described in the Table 5.2.

- MA-based SL orders verification mode. In this case the SL order is verified by the MA. The MA's period is defined by user and the *buy* and *sell* rules are described in the Table 5.3.

Signal	TP order rule
buy	IF $([price(t).level^{high} - \theta] > [price(t) + tpMin])$ THEN $(TP = price(t).level^{high} - \theta)$ ELSE (!valid trade)
sell	IF $([price(t).level^{low} + \theta] < [price(t) - tpMin])$ THEN $(TP = price(t).level^{low} + \theta)$ ELSE (!valid trade)

Table 5.1: Fibonacci and Pivot based TP orders verification modes. $price(t)$ is the current 'open' price; $TP(t)$ is the evolved size of a TP order from the GP individual; $TP(t).level^{low}$ is the nearest Fibo (Pivot) level below $TP(t)$; $TP(t).level^{high}$ is the nearest Fibo (Pivot) level above $TP(t)$; θ is the TP order threshold.

Signal	SL order rule
buy	IF $([price(t) - SL] < [price(t).level^{low} - \theta])$ THEN $(SL = price(t).level^{low} - \theta)$ ELSE (!valid trade)
sell	IF $([price(t) + SL] > [price(t).level^{high} + \theta])$ THEN $(SL = price(t).level^{high} + \theta)$ ELSE (!valid trade)

Table 5.2: Fibo and Pivot based SL orders verification modes. $price(t)$ is the current 'open' price; SL is the evolved size of a SL order from the GP individual; $price(t).level^{low}$ is the nearest Fibo (Pivot) level below $price(t)$; $price(t).level^{high}$ is the nearest Fibo (Pivot) level above $price(t)$; θ is the SL order threshold.

Signal	SL order rule
buy	IF $((price(t) > MA(t))$ AND $([price(t) - SL] < [MA(t) - \theta]))$ THEN $(SL = MA(t) - \theta)$ ELSE (!valid trade)
sell	IF $((price(t) < MA(t))$ AND $([price(t) + SL] > [MA(t) + \theta]))$ THEN $(SL = MA(t) + \theta)$ ELSE (!valid trade)

Table 5.3: MA based SL orders verification mode. An additional test is inserted to check that the price is on the relevant side of the moving average. $MA(t)$ is the scalar moving average at time step 't' as estimated by equation (5.1); SL is the evolved size of a SL order from the GP individual; θ is the SL order threshold.

5.5 Experimental Setup

The experimental setup is described below. In all cases FXGP employs the single trading agent mode and uses the same data set, currency pair (EURUSD), and the same parametrization. All parameters were optimized for the 2009 historical rates, as in Section 3.1.6. Please see Table 5.4 for summary of the parameters. To establish the bases for comparison, the trading activity for the same period Jan. 2010 — Nov. 2012 was simulated[1].

Parameter	Value
TIp (Minimum TI population size)	100
TIs (Maximum TI program size, steps)	8
Regs (Number of TI program registers)	2
DTp (DT population size)	100
DTgap (Number of DTs replaced in each generation)	25
DTmut (Relative probability of DT or TI mutation)	0.5
DTs (Maximum DT size, nodes)	6
Gmax (Maximum number of generations)	1000
Nt (Training partition size)	1000
Nv (Validation partition size)	500
SLmax (Maximum SL order size, pips)	100
SLmin (Minimum SL order size, pips)	5
TPmax (Maximum TP order size, pips)	300
TPmin (Minimum TP order size, pips)	30
τ (Training plateau length, generations)	200
α_v (DT-TI validation fraction)	0.95
Lrow (Maximum number of consequent losses)	3
Dd (Maximum drawdown, pips)	400
Hrow (Maximum number of consequent candlesticks without trading activity)	72

Table 5.4: Main FXGP parameters.

A total of five configurations are considered. The first configuration is the FXGP without trading orders verification (SL orders are evolved from training data). The second and third configurations use Fibonacci retracement in which two forms for the min–max definition are considered. The motivation behind these two configurations are that the close for a given candle statistic is more robust than the high/low price

---

[1] All runs were performed on a 2.8 GHz iMac computer with Intel Core i7 CPU, 16GB RAM and Mac OS X 10.7.2.

swing. The remaining configurations assume MA and Pivot TIs respectively. The configurations involve particular thresholds and associated rules:

1. **FXGP mode:** Unmodified version of FXGP, hence SL orders are limited to a simplistic threshold comparison.

2. **FXGPF*hl* mode:** The SL orders are verified by the Fibo levels (Table 5.2). The 0 and 100 Fibo levels are set to the recent significant Swing **Low** and Swing **High** prices (Figure 5.2). The TP order is placed 15 pips below the 161.8% level ("buy" signal) or 15 pips above the -61.8% level ("sell" signal). If the difference between the trading order ("buy" or "sell") price and the TP order is less than $tpMin$, then the trading signal is ignored and trading order is not opened.

3. **FXGPF mode:** The SL orders are verified by the Fibo levels (Table 5.2). The 0 and 100 Fibo levels are set to the recent significant Swing **Close** prices (Figure 5.4). The TP order is placed 15 pips below the 161.8% level ("buy" signal) or 15 pips above the -61.8% level ("sell" signal). If the difference between the trading order ("buy" or "sell") price and the TP order is less than the threshold $tpMin$, then the trading signal is ignored and trading order is not opened.

4. **Pivot mode:** The SL orders are verified by the Pivot levels (Table 5.2). The TP order is placed 15 pips below the R3 resistance level ("buy" signal) or 15 pips above the S3 support level ("sell" signal). If the difference between the trading order ("buy" or "sell") price and the TP order is less than $tpMin$, trading signal is ignored and trading order is not opened.

5. **MA mode:** The SL orders are verified by the MA (Table 5.3). The MA periods are set to 48 candlesticks (MA48), 72 candlesticks (MA72) or 96 candlesticks (MA96). TP orders performed significantly worse, so for clarity they are not reported here.

Figure 5.4: Fibonacci retracement. The 0 and 100 levels are drawn through recent significant Swing Close prices. Image produced using the MetaTrader 4 Forex trading platform `https://www.metatrader4.com`

5.6 Results

The results of all experiments are summarized in Table 5.5. Each experiment includes 100 simulations. Results are ranked in terms of the median number of pips[2] accumulated at the conclusion of the trading period. A Student's t-test (Table 5.6) confirms the independence of the distributions relative to the top ranked configuration. The *p-values* of Shapiro-Wilk test of normality for algorithm modes (Table 5.5) are shown in the Table 5.7. It is clear that validation of buy/sell orders using specific configurations of MA and Fibonacci derived levels was more effective than any other scenario. Moreover, MA is more sensitive to the specific parameterization assumed for the length of the moving average calculation. Conversely, the only design decision playing a role in the definition for the Fibonacci levels is the statistic, i.e., either close or high/low used to configure the 0 and 100 percent levels (compare Figure 5.2 with Figure 5.4). Employing the 'swing close' prices from the summary statistic of the

---

[2] Under the EURUSD market, a move of 0.0001 is equivalent to one pip.

candle is typically more robust than swing high or swing low prices. It is also clear that basing the TI on levels identified by pivot points is universally ineffective, albeit with respect to this currency pair and period.

The average number of trades per run and SL orders' statistics are summarized in the Table 5.8 for each mode in Section 5.5. Both of the top two TI configurations — FXGPF and MA72 — perform more valid trades than any other configuration other than Pivot based TI. However, the number of *triggered* stop-losses (as a fraction of the number of trades) under FGPF is significantly lower than any other configuration. This implies that the rules generated from this parameterization relied less on the 'corrective' effect of SL orders. Moreover, the Fibonacci levels based on the close prices were able to filter more of the noise effects than Fibonacci levels initialized under high–low prices. All adaptive schemes for validating SL orders resulted in a smaller SL order than the original SL thresholding scheme (FXGP) and therefore lost less through SL orders.

The champion configuration FXGPF was deployed as an ensemble of 3 trading agents (FXGPFT 3) and the results of 100 simulations were compared with the version without verification (FXGPT 3). The results are summarized in the Table 5.9. Comparison with Table 5.5 demonstrates that the ensemble version pushes the tail of the distribution up in each case. This makes the resulting trading agents less sensitive to initial conditions (more dependable) as, for example, 98 percent of the FXGPFT 3 runs now avoid recording a loss over the trading period. Figure 5.5 illustrates this phenomena in terms of a combined violin/box plot summarizing the distribution of cumulative pips for all 100 runs for each of the pairs of trading agent. Figure 5.6 provides an illustration of the cost of training/retraining under single and ensemble frameworks. Given that trading information characterizes 1 hour intervals, one can note that both forms of the algorithm operate in 'real-time'.[3] And finally, Figure 5.7 shows the distribution of the first time occurrences of the drawdown over trading simulation periods.

A final experiment is performed with the winning FXGPF configuration. Previous practice in the academic literature has been to describe performance of an investment strategy as the cumulative profit/loss collected over a total investment period. This

---

[3] The General Central Dispatch utility available as part of the Apple OS is used to schedule the execution of multiple threads during the ensemble experiments.

Algorithm's mode	Profitable runs (%)	Score (pips)				
		min	1st quartile	median	3rd quartile	max
FXGPF†	96	-993	1145	1847	2633	4474
MA72	90	-1193	766	1420	2121	3492
FXGPF*hl*	80	-2217	131	1190	1843	4146
FXGP†	74	-3154	-95	989	1918	4055
MA96	94	-1791	329	828	1647	3529
MA48	71	-2541	-208	534	1334	2827
Pivot	49	-3281	-715	-66	396	2263

Table 5.5: Single trading agent comparison. Results are sorted with respect to the median scores. † indicates the runs that are illustrated by the distribution of Figures 5.5 and 5.6.

FXGPF vs MA72	FXGPF vs FXGPF*hl*	FXGPF vs FXGP	FXGPF vs MA96	FXGPF vs MA48	FXGPF vs Pivot
5.56×10^{-3}	2.54×10^{-7}	9.00×10^{-6}	5.78×10^{-9}	8.29×10^{-16}	9.00×10^{-25}

Table 5.6: *p-values* for pairwise Student t-test.

FXGPF	MA72	FXGPF*hl*	FXGP	MA96	MA48	Pivot
0.9293	0.4411	0.2712	0.5832	0.0635	0.3890	0.0182

Table 5.7: Shapiro-Wilk test of normality, *p-values* for algorithm modes (Table 5.5)

Description	FXGPF	MA72	FXGPF*hl*	FXGP	MA96	MA48	Pivot
Average # of trades, per run	558	440	422	428	412	438	4845
Average # of triggered SL, per run	204	208	199	194	195	242	252
Average % of of triggered SL, per run	37	47	47	45	47	55	52
Average SL order size, pips	44	39	44	73	40	36	38

Table 5.8: SL algorithms comparison. Average number of trades and SL statistics.

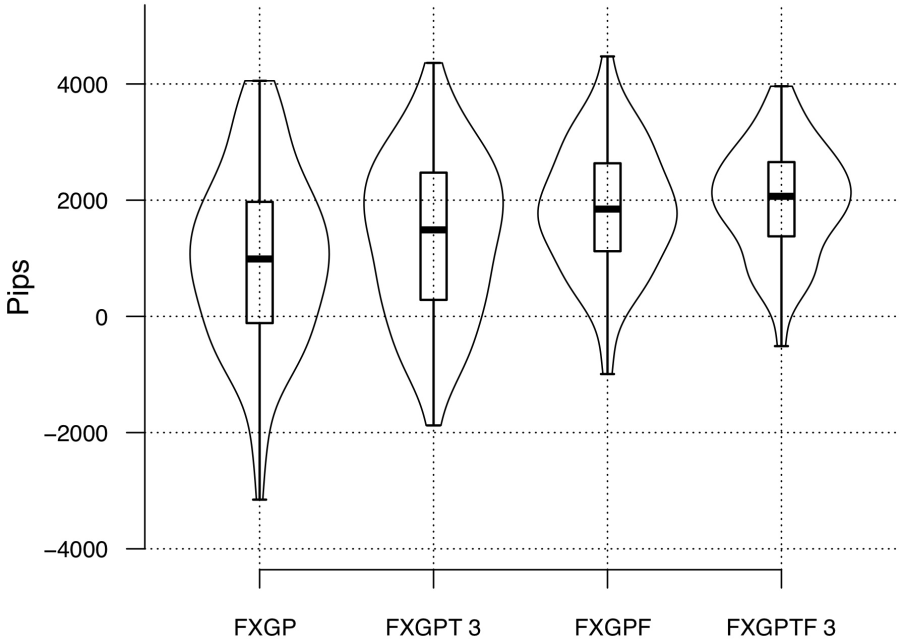


Figure 5.5: Distribution of cumulative scores in pips over trading simulation period of time for single (FXGP and FXGPF) and ensemble of 3 champion trading agents.

SL Algorithm	Profitable runs (%)	Score (pips) min	1st quartile	median	3rd quartile	max
FXGPFT 3†	98	-512	1383	2067	2648	3961
FXGPT 3†	81	-1877	289	1489	2463	4362

Table 5.9: Ensemble trading agent comparison for $k = 3$. Results are sorted with respect to the median scores. † indicates the runs that are illustrated by the distribution of Figures 5.5 and 5.6.

means that early successes can potentially mask latter losses. See for example, the first four rows of Table 5.10. With this in mind a trader actually using FXGP in practice might periodically revisit the parameterization using historical data. Thus, the final row of Table 5.10 illustrates the effectiveness of re-parameterizing FXGP using the data from 2012 and deploying this during 2013.

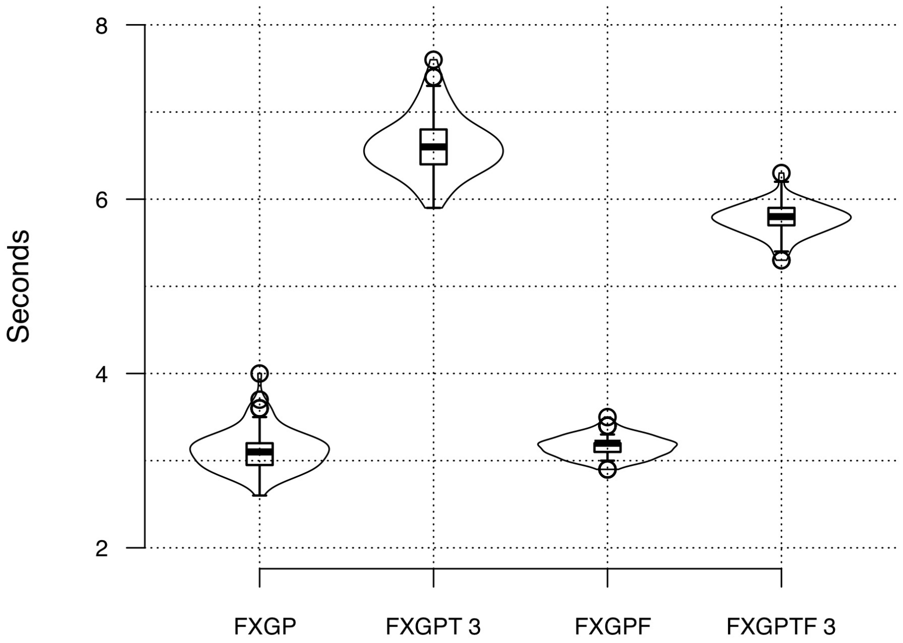


Figure 5.6: Distribution of average training times (over a run) per population (FXGP and FXGPF) and per team of three populations (FXGPT 3 and FXGPTF 3).

Year	Profitable runs (%)	Score (pips) min	1st quartile	median	3rd quartile	max
2010	100	223	1166	1672	2168	3570
2011	84	-979	178	675	1201	3127
2012	0	-1842	-1204	-947	-680	-59
2013	25	-1048	-430	-180	-4	481
2013†	81	-628	84	464	755	1758

Table 5.10: FXGPF annual results. 2013 vanilla and 2013 optimized† lines include the first 10 months of the year 2013. 2013 optimized† line shows result after optimization of the algorithm parameters based on the 2012 year historical rates.

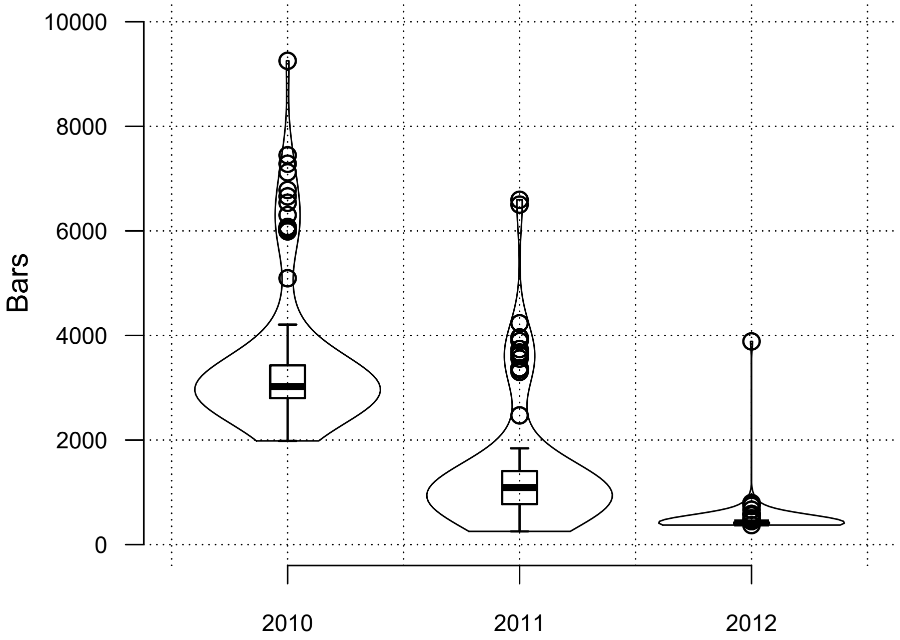


Figure 5.7: Distribution of the first time occurrences of the drawdown over trading simulation periods of time starting from January 2010 (2010), January 2011 (2011) and January 2012 (2012). 500 candlesticks period is approximately equivalent to a month.

5.7 Discussion

This section examined the usage of three different techniques for identifying retracement opportunities in the FXGP framework, namely moving average, pivot points, and Fibonacci ratios. The strategies based on the Fibonacci series clearly had the best performance of the three techniques. Moreover, Fibonacci retracement with 0% and 100% levels drawn through the Swing Close prices (FXGPF) is more efficient than Fibonacci retracement with 0% and 100% levels drawn through the Swing Low and Swing High prices for FXGP. The use of Fibonacci retracement to define the SL order position increased the median score by 86.7%, increased the percent of profitable trades by 29.7% (Table 5.5), and reduced the average size of a SL order by 51% (Table 5.8) compared to the FXGP without orders verification. The use of Fibonacci retracement to define the SL order position reduced the retraining time of the team of three trading agents (Figure 5.6) by 12%. This reduction likely occurs because the use of Fibonacci levels to define the SL order position increases chances of a DT–TI population passing the validation process. Of the different modes of FXGP, FXGPF had the biggest average number of trades per simulated period of time and the lowest percent of triggered SL orders (Table 5.8). Teams of trading agents were confirmed to be more likely to return profitable trading strategies than single trading agents (Table 5.9). Note, however, that these results are likely to be sensitive to the trading conditions, trading assets and time period(s). Specifically, it is anticipated that the floating Bid-Ask spread will significantly affect the performance of automated trading agents, as was shown in Section 3.3 under the case of Foreign Exchange markets.

Chapter 6

Asset Selection Algorithm for Frequent Intraday Trading

Portfolio optimization represents an attempt to distribute investment (funds) across a set (or subset) of available assets to achieve a tradeoff between risk and return. While there are many fundamental works and novel research on how to select a *long term portfolio* of trading assets and distribute funds within it (Section 2.5), it is apparent that there is a lack of research that answers the same question in case of frequent intraday trading. Attempts to apply traditional methods of long term portfolio selection to intraday trading on short time intervals[1] (hereafter frequent trading) may not be optimal since the variation of the price is different and transaction costs are potentially a much higher proportion of the cost of trading (Section 2.5). This chapter attempts to answer the question of how an investor should select the subset of trading assets for intraday trading and how to distribute funds across the selected trading assets. To do so, the NASDAQ 100 historical rates are used. The NASDAQ 100 Index is a set of most actively traded companies listed on the NASDAQ stock exchange [69]. Moreover, data from the NASDAQ exchange has previously been observed to have higher levels of volatility than from the NYSE [73].

In light of the above goals, the FXGP algorithm (Section 3.3) will be deployed to estimate the return for *each stock independently* in a ‘bag’ of 86 stocks (identified in Section 6.1). The returns are ranked, and the top one or more stocks traded during the following day (while simultaneously simulating the returns for all available stocks). Two simple ranking criterion/ heuristics are investigated for defining the intraday portfolio: Moving average of daily returns and Moving Sharpe ratio. Their performance was compared to the full portfolio (86 stocks), random stock selection and Kelly criterion [86, 26] to find optimal intraday portfolio size (for the proposed configuration). Unlike the earlier sections, FXGP is now deployed with respect to price data for stocks (not foreign exchange currency data). Moreover, FXGP trading

---

[1] Up to a minute, but no less than a second.

decisions are now used to potentially develop a policy for multiple stocks simultaneously.

6.1 Properties of the Stock Selection Dataset

The NASDAQ 100 historical rates ('Mid' prices) for the period from August 1, 2014, to August 31, 2017, are used in this Chapter. All stocks with missing or out of range date and time stamps and stocks with missing, '0' or negative prices were excluded. The remaining stock data is then preprocessed into a candlestick representation assuming a 1-minute duration with each candle expressing: open-high-low-close price information for the 1-minute interval with a total of 390 candlesticks or 6.5 hours per trading day, i.e. ≈ 280800 candlesticks *per stock* over the 3 year interval.

The resulting portfolio includes the following 86 NASDAQ 100 stocks: AAL, AAPL, ADBE, ADI, ADP, ADSK, AKAM, ALXN, AMAT, AMGN, AMZN, ATVI, BIDU, BIIB, BMRN, CA, CELG, CERN, CHKP, CMCSA, CSCO, CSX, CTAS, CTRP, CTSH, CTXS, DISCK, DISH, DLTR, EA, EBAY, ESRX, FAST, FB, FOX, FOXA, GILD, HAS, HOLX, HSIC, IDXX, ILMN, INCY, INTC, INTU, ISRG, JBHT, KLAC, LBTYK, LRCX, MAR, MAT, MCHP, MDLZ, MSFT, MU, MXIM, NCLH, NFLX, NTES, NVDA, ORLY, PAYX, PCAR, PCLN, QCOM, ROST, SBUX, SHPG, SIRI, STX, SWKS, SYMC, TSCO, TSLA, TXN, ULTA, VIAB, VOD, VRSK, VRTX, WBA, WDC, WYNN, XLNX, XRAY. This work assumes that all 86 stocks stay in NASDAQ 100 and do not split during the simulated period of time.

The annual Skewness and Kurtosis metrics of all 86 stocks are listed in the Table A.1, and the distributions are shown in Figures 6.1 and 6.2, respectively. In short, the trends as captured by the Skewness and Kurtosis metrics differ significantly on a year-to-year basis. During 2015 there are 17 stocks with a Kurtosis greater than 3,[2] whereas in 2016 there are 23 such stocks of which only 6 are common with the stocks observing this characteristic in 2015. In 2017 there are 18 such stocks, of which 5 are common with those in the previous year. Moreover, as captured by the violin plots of Figures 6.1 and 6.2, there is a significant difference in the magnitudes of the metrics, particularly between 2015 and 2016/17. Finally, it is worth noting that there is also a lack of continuity in the Skewness statistic. For example, there is little correlation

---

[2] Indicates a frequency of outliers above that experienced by a normal distribution.

in the sign of Skewness between consecutive years of the same stock.

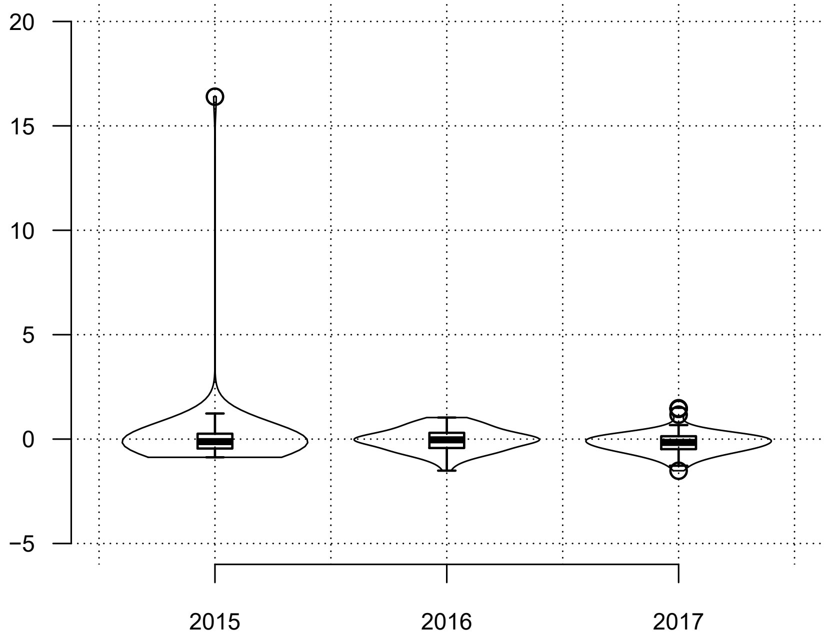


Figure 6.1: Distribution of annual Skewness of NASDAQ 100 stocks.

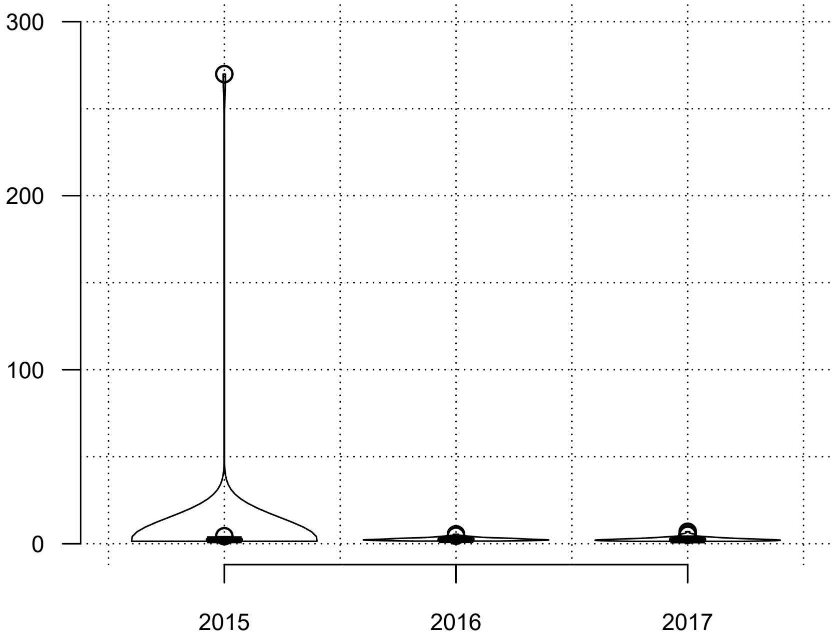


Figure 6.2: Distribution of annual Kurtosis of NASDAQ 100 stocks.

6.2 Framework

Figure 6.3 provides an overview to the stock selection framework. A cold start period provides data from which GP-trading agents are evolved for each of the N stocks in the portfolio. Stock data is described in terms of 1-minute candlesticks. Each GP agent learns to maximize their respective stock's return independently by simultaneously designing the technical indicators and determining the buy-hold-sell signals. At the end of the trading day, the return from each agent is used to rank each stock from the portfolio, and the stocks with the highest S ranks are selected for trading at the next trading day (Section 6.3). During trading day $t + 1$, the S agents selected to make investments, trade with a money management policy, whereas the remaining GP agents continue to trade under simulated conditions.

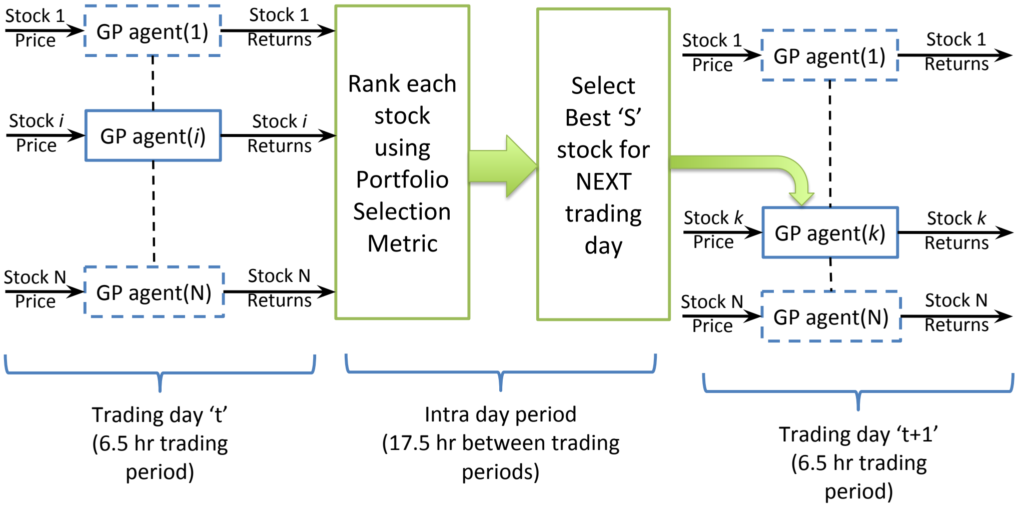


Figure 6.3: Overall framework for intraday stock selection and trading. During trading day, t, all agents estimate their return. These returns are then ranked using the Portfolio Selection Metric. The best stocks are then traded in the following trading period, $t + 1$. Solid boxes indicate agents selected to perform intraday trading (part of the 'portfolio') in any particular period, with the remaining agents simulating their return.

6.3 Stock Selection Algorithms

Three intraday portfolio selection algorithms are examined: Moving Average of Daily Profits (MAR), Moving Sharpe Ratio (MS) and, in addition, the Kelly Criterion (Kelly) which is used as a comparator mechanism for stock selection where the metric defines a probabilistic formalism for stock selection [86]. Thus, the evaluated stocks are sorted in descending order according to the corresponding metric: MAR, MS or Kelly (hereafter the 'weight') and the best S stocks[3] are used for intraday trading.

A trading account balance is distributed among the stocks in proportion to their weights. For example, a ranking by the Moving Sharpe Ratio for the top $S = 3$ stocks

---

[3] S is a number of stocks that can be traded in the daily portfolio (Figure 6.3).

returns metric values of 0.9, 0.6, 0.1 respectively. The 1st stock will see $0.9 \div (0.9 + 0.6 + 0.1)$ or 0.5625 of the account invested in this stock (a trading share), the 2nd stock will see a weight of $0.6 \div (0.9 + 0.6 + 0.1)$ or 0.375 of the account invested and the 3rd stock will receive $0.1 \div (0.9 + 0.6 + 0.1)$ or 0.0625 of the account balance invested.

6.3.1 Moving Average of Daily Returns

This research will adopt the following moving average of a stock's daily returns (profits or losses) for any day i, calculated as follows:

$$MAR_i = \frac{\sum_{j=1}^{d} P_j}{d} \tag{6.1}$$

where d is the MAR period (number of previous daily returns) and P_j is a daily return of a day j. Such a metric implicitly accounts for profit and loss as the MAR is estimated over a sequential set of daily returns

6.3.2 Moving Sharpe Ratio

The Sharpe Ratio attempts to quantify the likelihood of seeing a return in an investment relative to its risk. Thus, given the Sharpe Ratio for two assets, the asset with the higher Sharpe Ratio is more likely to result in an excess return. A Moving Sharpe ratio is a Sharpe Ratio that is calculated based on previous daily returns as follows [93]:

$$MS_i = \frac{(R_{mean} - R_{free})}{R_{deviation}} \tag{6.2}$$

where MS_i is the Sharpe ratio on a day i, R_{mean} is the mean portfolio return over the d previous days, $R_{deviation}$ is the standard deviation of d previous daily returns and R_{free} is the best available risk-free rate of return. For the stock selection purpose, the relative value of the Sharpe Ratio of each stock is important. Hence, this work assumes that R_{free} is equal to 0.

6.3.3 The Kelly Criterion

The Kelly Criterion represents a strategy for optimal 'bet sizing' for higher returns over repeated trials, i.e. maximizing the expected logarithm of wealth. As such the Kelly Criterion has seen use for classical investment decision making [86], but also shares similar drawbacks to MVO, such as the need to have reliable estimates of the forecasted probability distributions. The Kelly Criterion is calculated as follows:

$$K_i = W - \frac{1 - W}{R} \tag{6.3}$$

where W is a winning probability and R as a win/loss ratio. The W and R are calculated based on the returns of all trades for the last d days [86].

6.4 Trading Conditions and Experimental Methodology

The following trading conditions/approach to money management were assumed: Initial account balance \$10,000,000, with a flat rate \$5 per trade and fixed spread \$0.01. Only 10% of the account balance at the beginning of each day is used to trade stocks. To provide 'Bid' and 'Ask' prices with the fixed spread \$0.005 was subtracted/added from/to downloaded 'Mid' prices.

A total of 50 sets of simulation runs are performed where each set includes trading signals for all 86 stocks and covers 2015–2017 years. The MAR, MS, and Kelly (Section 6.3) were calculated for 3, 5, 10, 20 and 40 days, where different estimation periods might impact on the quality of the investment strategy (e.g. under low frequency strategies, longer estimation periods are recommended for larger portfolios [38]). Four different intraday portfolio sizes were investigated: $S = \{1, 3, 5, 10\}$ stocks for MAR, MS, Kelly and the random stock selection or 86 stocks for the full portfolio.

In the following, a baseline for the case of no GP trading agents (buy-and-hold strategy, full portfolio, and random stock selection) was developed. The framework of Section 6.2, Figure 6.3 is then adopted with each of the three ranking metrics: Kelly Criterion, Moving Average of Daily Returns, and Moving Sharpe Ratio.

In all cases results are expressed in terms of the average annual performance from 50 simulation runs using the following performance metrics:

- Profit (%) — the difference between opening account balance ($10,000,000$ USD, Section 6.4) and balance at the end of the period including any transaction costs.
- Sharpe ratio (monthly returns) — defined using Equation (6.2) with R_{mean} and $R_{deviation}$ estimated monthly over the *entire* year (12 months).
- Hit rate (%) — attempts to express the accuracy of buy/sell signals relative to the outcome of the following candlestick. Thus, if the GP trading agent issued a buy (sell) signal and the price increased (decreased) at the next candlestick, this would be considered a 'hit'.
- Average profit per trade (USD) — calculated as a sum of all trades within a year with a positive value divided by the number of such trades.
- Average loss per trade (USD) — calculated as a sum of all trades within a year with a negative value divided by the number of such trades.
- Number of trades — is the raw count of the number of trades placed by the GP trading agents in total.

6.5 Results

6.5.1 Buy-and-Hold Strategy

The Buy-and-Hold investment strategy for all 86 stocks in the portfolio[4] results in the investment outcomes of Table 6.1. Exceeding the performance of a Buy-and-Hold policy is not a forgone conclusion, with previous research employing a GP trading agent failing to better the Buy-and-Hold strategy, e.g. [29, 3, 134]. Other recent research using deep learning approaches also rely solely on Buy-and-Hold baselines [12]. One reason for use of the Buy-and-Hold baseline is that the strategy explicitly minimizes the transaction cost. Moreover, as noted in a recent survey article of GP in finance and economics [20], positive economic development will tend to benefit a Buy-and-Hold strategy, where this is apparent in the increasing Profits recorded in Table 6.1 for 2016 and 2017. In addition, it can be noted that as the interval describing price movements becomes smaller, it becomes increasingly difficult to identify underlying

---

[4] All stocks from the portfolio are held, as there is no basis for selecting a subset of specific stock.

trends as the price signal becomes more unpredictable. All three of these factors (transaction fee, Nasdaq was flat in 2015, but gained considerable value in 2016 and 2017,[5] TI defined over 1-minute interval) play to the advantages of a Buy-and-Hold strategy.

Investment period	Profit (%)
2015	0.83
2016	1.22
2017	1.74
2015 to 2017	4.41

Table 6.1: Buy-and-hold strategy investment outcomes.

6.5.2 Full Portfolio and Random Stock Selection

This section summarizes the results for the control cases in which there were *no* GP trading agents involved (Tables 6.2 — 6.4, one table per year). Specifically, the case of investing in a full $\frac{1}{N}$ portfolio of all 86 stocks is considered[6] and the case of the intraday portfolios in which 1, 3, 5 or 10 stocks are randomly selected from the full set of stocks. Figures 6.4 — 6.9 show the distribution of annual profits and Sharpe ratios over 50 runs for each of the three calendar years (2017 covers the first 8 months for which data is available). It is readily apparent that, even under the context of random stock selection, increasing the number of stock traded in the portfolio decreases the variance. However, it is also apparent that although variance decreases with increasing portfolio size, the average return decreases. This implies that some subset of stock, for some subset of the 50 runs, can be profitable (indeed the top quartile for the $S = 1$ portfolios in all three years can be profitable/have a positive Sharpe Ratio). However, as the number of trades are also growing with a number of stocks in a daily portfolio, then the commission paid is growing as well. The next step is to introduce an agent to perform the stock selection and trading.

---

[5] `https://www.macrotrends.net/2489/nasdaq-composite-index-10-year-daily-chart`

[6] Implies that each stock is invested in equally each day, subject to the 10% account balance and flat rate rules from Section 3.1.5.

Metric	Intraday portfolio, stocks				
	1	3	5	10	86
Profit (%)	-0.32	-0.73	-1.01	-2.11	-17.92
Sharpe ratio (monthly returns)	-0.03	-0.16	-0.28	-0.77	-3.61
Hit rate (%)	31.89	32.34	32.27	32.12	30.43
Average profit per trade (USD)	4982.53	1639.75	966.41	477.69	44.25
Average loss per trade (USD)	-2372.32	-802.26	-474.88	-241.07	-33.76
Number of trades	2103	6168	10538	20878	178926

Table 6.2: Full portfolio and random stock selection, 2015

Metric	Intraday portfolio, stocks				
	1	3	5	10	86
Profit (%)	0.33	-0.36	-0.8	-1.72	-17.17
Sharpe ratio (monthly returns)	0.06	-0.1	-0.25	-0.65	-3.23
Hit rate (%)	32.3	31.7	31.38	31.47	28.75
Average profit per trade (USD)	5037.48	1640.9	972.78	481.12	46.56
Average loss per trade (USD)	-2401.59	-772.16	-456.95	-233.52	-32.67
Number of trades	1979	6044	10107	20154	90196

Table 6.3: Full portfolio and random stock selection, 2016

Metric	Intraday portfolio, stocks				
	1	3	5	10	86
Profit (%)	-0.33	-0.16	-0.49	-1.13	-10.5
Sharpe ratio (monthly returns)	-0.07	-0.06	-0.29	-0.93	-4.22
Hit rate (%)	30.94	30.89	30.53	30.56	25.6
Average profit per trade (USD)	4119.55	1364.82	816.73	399.53	42.04
Average loss per trade (USD)	-1908.05	-619.26	-371.51	-189.22	-27.77
Number of trades	1235	3670	6163	12288	105966

Table 6.4: Full portfolio and random stock selection, 2017

6.5.3 The Kelly Criterion

This section summarizes results of stock selection based on: 1) GP trading agent estimation of the daily returns for each of the 86 stock and, 2) ranking of agent returns for stock selection using the Kelly Criterion (Section 6.3.3). As can be seen

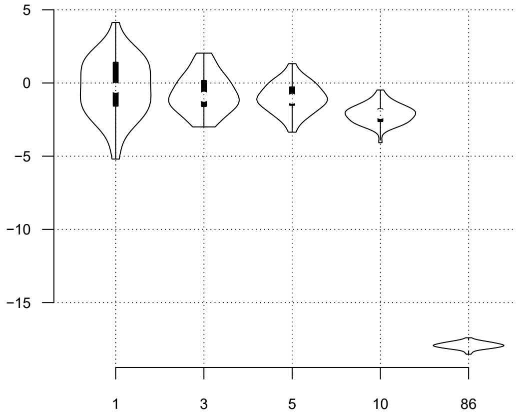


Figure 6.4: Full portfolio and random stock selection (2015): Profits (%) distribution

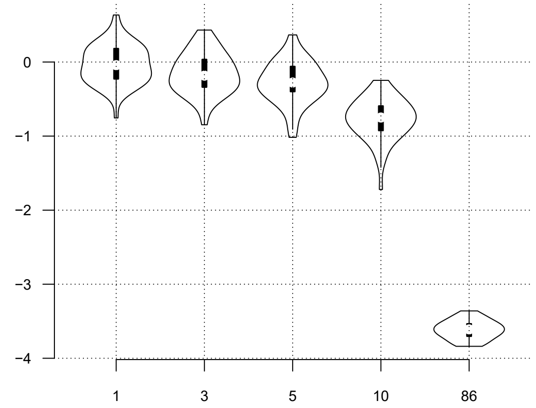


Figure 6.5: Full portfolio and random stock selection (2015): Sharpe ratios distribution

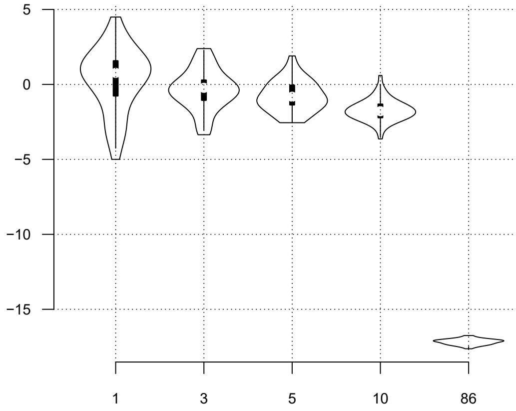


Figure 6.6: Full portfolio and random stock selection (2016): Profits (%) distribution

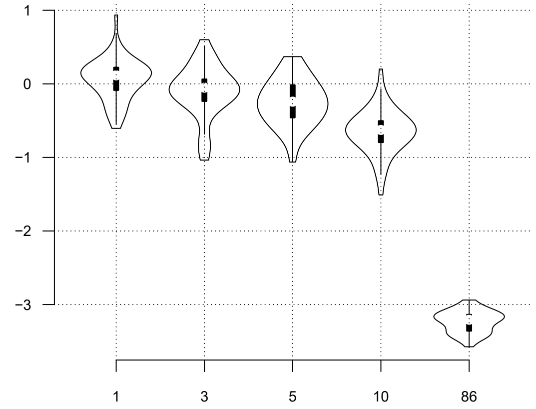


Figure 6.7: Full portfolio and random stock selection (2016): Sharpe ratios distribution

from Table 6.5 and Fig. 6.14, only two cases were profitable. In both cases, a small profit is returned (0.1% and 0.15%) over the eight months of 2017 and only in the case of trading one stock per day. Indeed, the visualization of portfolio size (S) and estimation periods for the ranking metric are generally not profitable (Fig. 6.10 through Fig. 6.15), with the dependent (z-axis) variable dominated by negative values. In general, with an intraday portfolio size of 1 to 10 stocks, this method gives almost the same results as random stock selection from Section 6.5.2, and fails to approach the return of the Buy-and-hold strategy (Table 6.1).

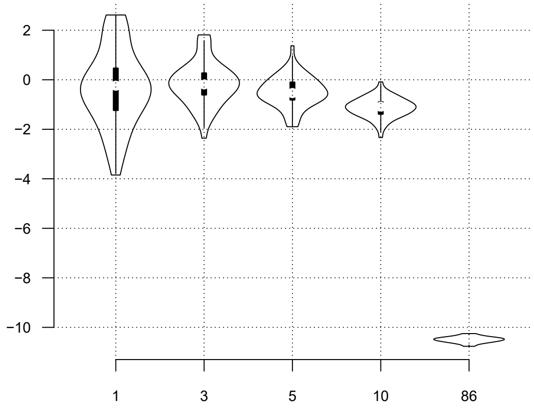


Figure 6.8: Full portfolio and random stock selection (2017): Profits (%) distribution

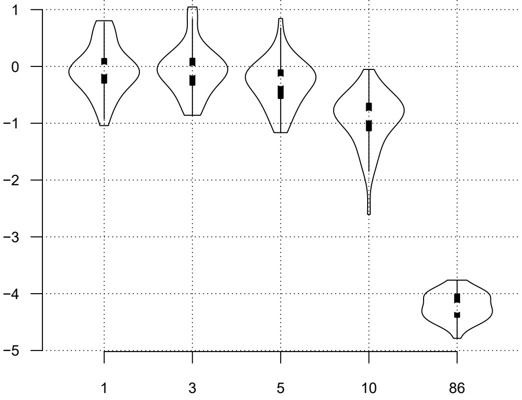


Figure 6.9: Full portfolio and random stock selection (2017): Sharpe ratios distribution

Year	Period, days	Intraday portfolio, stocks			
		1	3	5	10
2015	3	-0.97	-1.74	-1.87	-2.13
	5	-1.56	-1.62	-1.60	-1.90
	10	-1.22	-1.25	-1.33	-1.75
	20	**-0.75**	-1.31	-1.41	-1.87
	40	-1.70	-1.54	-1.76	-1.99
2016	3	-1.20	-1.02	-1.05	-1.19
	5	-0.79	-0.97	-0.90	-1.15
	10	-1.06	-0.85	-0.77	-1.03
	20	-0.87	**-0.63**	-0.74	-1.15
	40	-1.53	-1.03	-0.79	-1.26
2017	3	-0.20	-0.12	-0.24	-0.57
	5	-0.17	-0.20	-0.31	-0.56
	10	-0.16	-0.22	-0.47	-0.67
	20	0.10	-0.31	-0.41	-0.72
	40	**0.15**	-0.25	-0.34	-0.68

Table 6.5: Kelly Criterion stock selection: Profit, %

6.5.4 MAR of Daily Profits

Stock selection will now be formed using GP trading agent estimation of the daily returns for each of the 86 stocks followed by ranking of agent returns for stock selection using the MAR of daily profits. Profitability is now demonstrated in all cases when

Year	Period, days	Intraday portfolio, stocks			
		1	3	5	10
2015	3	-0.13	-0.41	-0.52	-0.69
	5	-0.22	-0.39	-0.47	-0.70
	10	-0.19	-0.30	-0.38	-0.65
	20	-0.13	-0.33	-0.45	-0.74
	40	-0.27	-0.43	-0.59	-0.84
2016	3	-0.16	-0.23	-0.29	-0.42
	5	-0.10	-0.21	-0.23	-0.37
	10	-0.15	-0.18	-0.19	-0.32
	20	-0.12	-0.14	-0.22	-0.37
	40	-0.22	-0.24	-0.21	-0.41
2017	3	-0.03	-0.07	-0.15	-0.49
	5	-0.06	-0.08	-0.20	-0.48
	10	-0.03	-0.09	-0.27	-0.55
	20	0.03	-0.15	-0.26	-0.61
	40	0.03	-0.14	-0.23	-0.57

Table 6.6: Kelly Criterion stock selection: Sharpe ratio

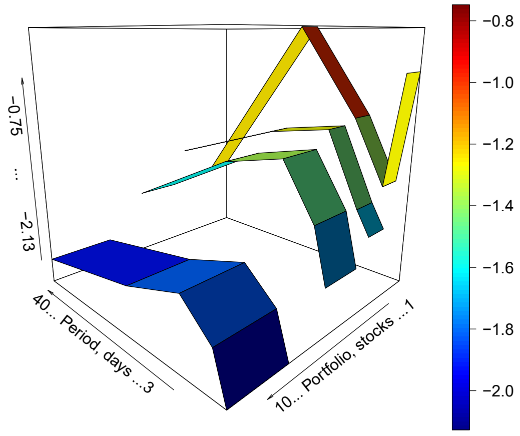


Figure 6.10: Kelly Criterion stock selection (2015): Profit, %

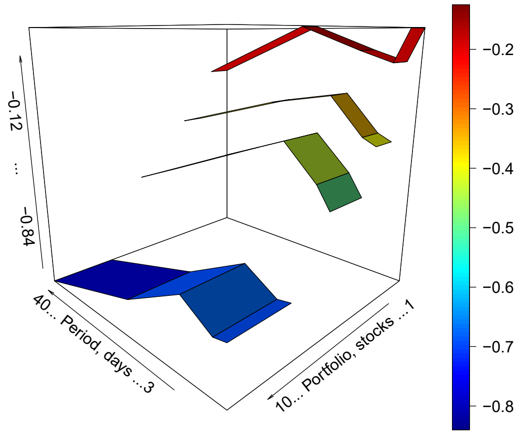


Figure 6.11: Kelly Criterion stock selection (2015): Sharpe ratio

the daily portfolio included five or less stocks (Table 6.11 and Fig. 6.16, 6.18, 6.20). In general, a smaller number of stocks per day and a longer estimation period (of

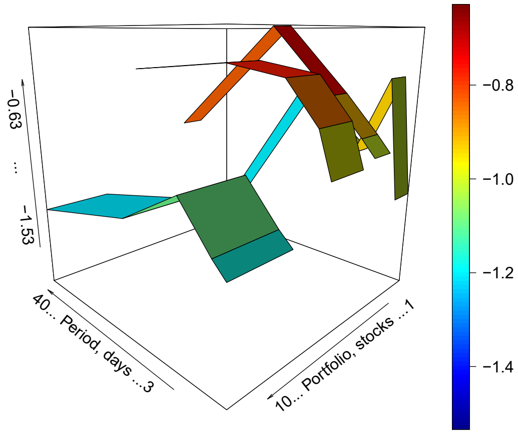


Figure 6.12: Kelly Criterion stock selection (2016): Profit, %

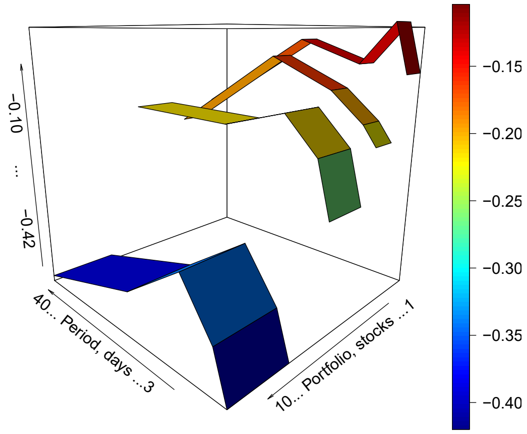


Figure 6.13: Kelly Criterion stock selection (2016): Sharpe ratio

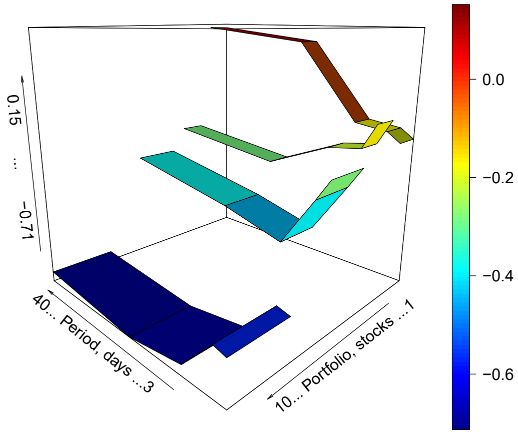


Figure 6.14: Kelly Criterion stock selection (2017): Profit, %

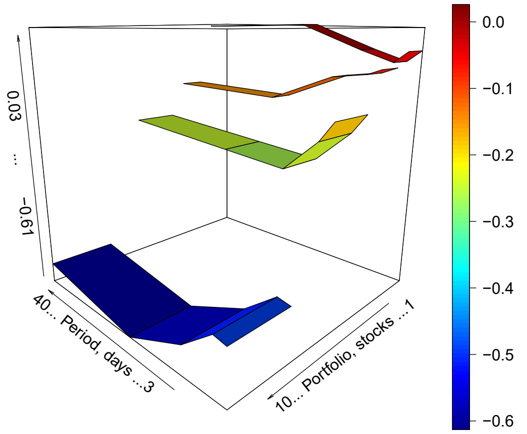


Figure 6.15: Kelly Criterion stock selection (2017): Sharpe ratio

the ranking statistic) lead to higher profit and better Sharpe ratio (Table 6.12 and Fig. 6.17, 6.19, 6.21). This tendency remains the same over all three years (2015 – 2017). At the same time, reducing the number of stocks per day improves the 'Hit ratio' (Table 6.13) and increasing of the MAR period from three to 40 days improves by more than 50% the 'Average profit' to 'Average loss' ratio (Tables 6.15 and 6.16).

Year	Period, days	Intraday portfolio, stocks			
		1	3	5	10
2015	3	44.04	43.21	43.03	42.42
	5	44.05	43.31	43.26	42.65
	10	44.30	43.85	43.34	41.78
	20	44.61	43.60	42.92	40.11
	40	43.61	42.56	40.96	35.98
2016	3	45.00	44.63	44.40	43.52
	5	45.50	44.72	44.56	43.03
	10	45.63	44.99	44.80	41.26
	20	44.89	44.50	42.12	32.96
	40	44.41	41.11	35.75	28.43
2017	3	45.91	45.56	45.19	44.03
	5	45.48	45.07	44.80	43.98
	10	44.78	44.83	44.34	43.55
	20	45.93	44.67	44.48	42.13
	40	46.47	45.22	44.45	36.20

Table 6.7: Kelly Criterion stock selection: Hit rate, %

Year	Period, days	Intraday portfolio, stocks			
		1	3	5	10
2015	3	432	1329	2315	5226
	5	411	1319	2332	5482
	10	412	1332	2394	6079
	20	411	1366	2547	8267
	40	411	1554	3414	12781
2016	3	421	1330	2324	5217
	5	406	1313	2300	5367
	10	405	1318	2327	6115
	20	416	1383	2700	11454
	40	427	1725	4889	24305
2017	3	291	874	1520	3350
	5	272	858	1495	3363
	10	270	865	1524	3443
	20	264	853	1506	3775
	40	255	833	1510	5901

Table 6.8: Kelly Criterion stock selection: Number of trades

Year	Period, days	Intraday portfolio, stocks			
		1	3	5	10
2015	3	8552.43	2758.13	1614.29	759.08
	5	8224.27	2724.21	1589.14	731.15
	10	8428.21	2747.19	1597.24	700.70
	20	8240.55	2663.91	1513.14	563.72
	40	8000.13	2473.70	1270.00	453.98
2016	3	7857.96	2643.22	1579.66	759.33
	5	8153.98	2679.84	1582.39	750.21
	10	8249.74	2688.74	1587.10	710.26
	20	8347.12	2690.88	1518.41	516.56
	40	8214.99	2479.46	1176.13	328.99
2017	3	6342.42	2158.50	1276.12	616.13
	5	6654.42	2201.49	1301.88	620.14
	10	6956.19	2239.10	1312.81	619.07
	20	6758.22	2189.52	1279.14	575.21
	40	6703.45	2188.26	1274.75	461.98

Table 6.9: Kelly Criterion stock selection: Average profit per trade, USD

Year	Period, days	Intraday portfolio, stocks			
		1	3	5	10
2015	3	-7128.20	-2328.02	-1361.44	-630.00
	5	-7147.78	-2299.01	-1332.75	-604.12
	10	-7235.54	-2314.08	-1321.73	-553.93
	20	-6973.77	-2230.61	-1239.60	-419.85
	40	-6921.48	-2021.24	-982.19	-284.47
2016	3	-6947.90	-2268.61	-1342.55	-625.66
	5	-7158.92	-2300.43	-1342.81	-604.76
	10	-7396.57	-2316.58	-1348.31	-531.18
	20	-7173.03	-2245.47	-1166.77	-273.74
	40	-7253.24	-1896.03	-721.50	-138.53
2017	3	-5505.78	-1831.85	-1080.93	-515.28
	5	-5649.78	-1847.54	-1093.04	-516.67
	10	-5730.33	-1862.79	-1100.00	-512.44
	20	-5659.02	-1833.23	-1073.86	-453.62
	40	-5694.05	-1860.65	-1060.48	-288.02

Table 6.10: Kelly Criterion stock selection: Average loss per trade, USD

Moreover, the profile of the portfolio size (S) and estimation period for the ranking metric now demonstrate a smooth increase/decrease as parameters are varied, hence, providing more certainty to parameter tuning, e.g. compare Profitability of Kelly Criterion in each year (Fig 6.10, 6.12, 6.14) with that under MAR (Fig 6.16, 6.18, 6.20).

Year	Period, days	Intraday portfolio, stocks			
		1	3	5	10
2015	3	5.94	4.31	2.60	-0.74
	5	7.98	5.48	3.42	-0.53
	10	14.33	9.19	5.60	0.45
	20	20.84	12.59	7.52	1.32
	40	**22.28**	15.29	9.30	2.71
2016	3	15.46	9.98	6.43	1.47
	5	23.94	14.64	9.55	3.11
	10	36.88	20.28	13.14	5.39
	20	45.89	25.04	16.26	7.63
	40	**47.41**	27.86	19.11	10.37
2017	3	4.26	2.43	1.38	-0.62
	5	5.21	3.20	1.85	-0.49
	10	7.73	4.67	2.55	-0.40
	20	10.16	6.18	3.40	0.19
	40	**12.76**	7.92	4.49	1.01

Table 6.11: MAR stock selection: Profit, %

6.5.5 Moving Sharpe Ratio

Stock selection will now be performed with GP trading agent estimation of the daily returns for each of the 86 stocks and the ranking of agent returns for stock selection using the Moving Sharpe ratio. Relative to the MAR case, the average profitability over each year increases in all parameterizations other than for the 3 day estimation period (Table 6.17 and Figures 6.17, 6.19, 6.21) and improves the Sharpe ratio in all but 5 of 60 parameterizations (Table 6.18 and Figures 6.23, 6.25, 6.27). Reducing the number of stocks per day improves the 'Hit ratio' (Table 6.19) and increasing of the estimation period (of the ranking statistic) from three to 40 days improves 'Average profit' to 'Average loss' ratio (Tables 6.21 and 6.22). Parameter sensitivity is again

Year	Period, days	Intraday portfolio, stocks			
		1	3	5	10
2015	3	0.43	0.54	0.43	-0.20
	5	0.56	0.64	0.52	-0.14
	10	1.01	1.14	0.89	0.11
	20	1.50	1.51	1.26	0.34
	40	1.92	1.93	1.60	0.71
2016	3	0.75	0.84	0.75	0.28
	5	0.99	1.11	1.03	0.53
	10	1.43	1.47	1.31	0.84
	20	2.02	1.80	1.53	1.05
	40	2.52	2.05	1.74	1.29
2017	3	0.80	0.75	0.62	-0.39
	5	0.96	1.13	0.92	-0.30
	10	1.26	1.31	1.06	-0.24
	20	1.76	1.69	1.34	0.12
	40	2.78	2.23	1.83	0.64

Table 6.12: MAR stock selection: Sharpe ratio

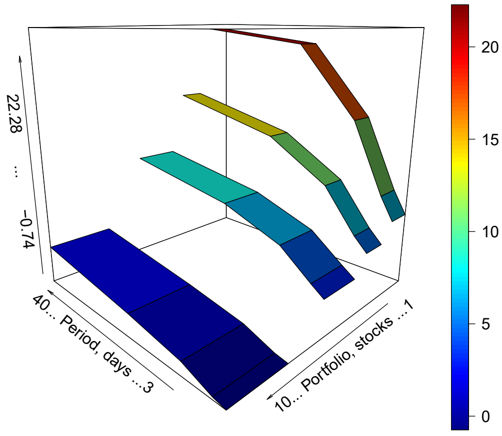


Figure 6.16: MAR stock selection (2015): Profit, %

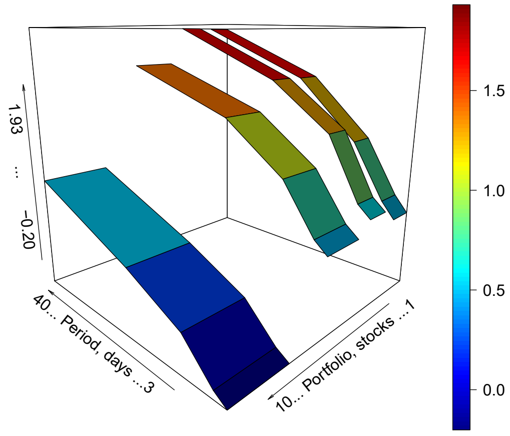


Figure 6.17: MAR stock selection (2015): Sharpe ratio

improved with much broader shoulders to the curves than appearing under MAR. In short, employing the Moving Sharpe ratio produces a better characterization of profit

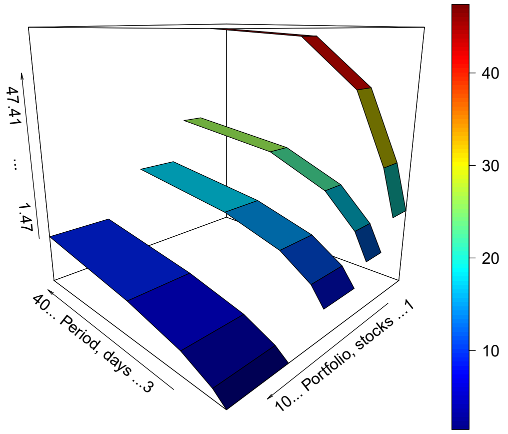


Figure 6.18: MAR stock selection (2016): Profit, %

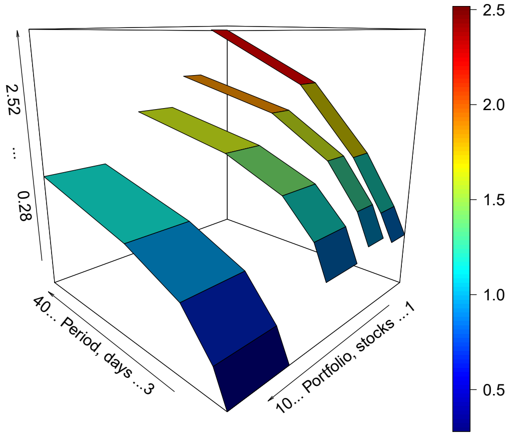


Figure 6.19: MAR stock selection (2016): Sharpe ratio

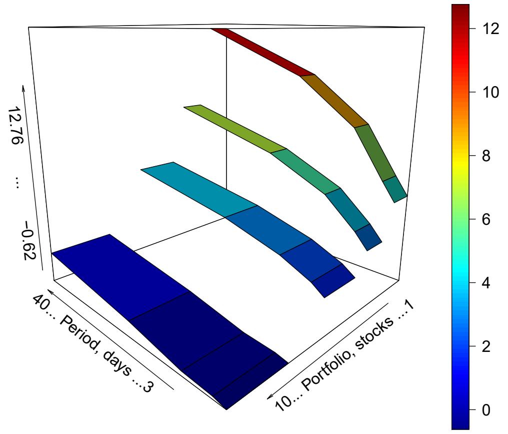


Figure 6.20: MAR stock selection (2017): Profit, %

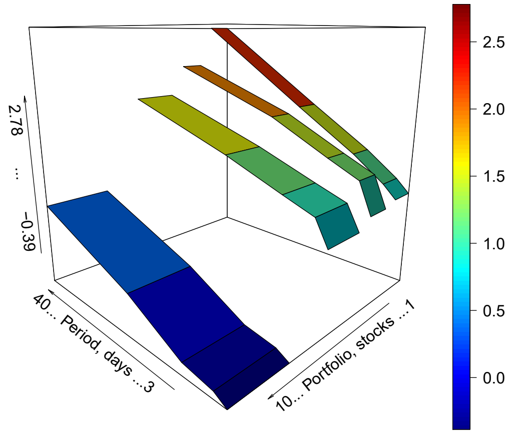


Figure 6.21: MAR stock selection (2017): Sharpe ratio

versus risk than available when using either Kelly Criterion or MAR.

Year	Period, days	Intraday portfolio, stocks			
		1	3	5	10
2015	3	30.71	28.91	28.43	28.10
	5	29.25	27.63	27.33	27.38
	10	27.04	26.02	26.10	26.67
	20	25.24	24.94	25.20	26.41
	40	24.47	24.04	25.03	26.55
2016	3	26.76	26.25	26.03	26.21
	5	25.88	25.25	25.21	25.73
	10	23.96	24.10	24.49	25.49
	20	22.50	23.70	24.32	25.55
	40	21.46	23.84	24.55	25.71
2017	3	27.16	26.53	26.25	26.01
	5	26.44	25.33	25.09	25.25
	10	25.53	24.07	23.90	24.45
	20	24.23	22.97	23.33	24.27
	40	22.16	22.54	23.22	24.33

Table 6.13: MAR stock selection: Hit rate, %

Year	Period, days	Intraday portfolio, stocks			
		1	3	5	10
2015	3	6324	19074	30134	52243
	5	8485	24585	37805	61635
	10	13506	36278	51279	72893
	20	19418	47915	63010	81293
	40	22969	58466	70882	87161
2016	3	12226	29178	41504	62184
	5	17585	38251	51362	70539
	10	27748	52332	63624	78732
	20	35998	62059	71073	83308
	40	40227	67347	75240	86969
2017	3	4150	11276	17316	30242
	5	5451	14916	22492	35899
	10	8174	22498	31198	43005
	20	11877	31578	38775	47720
	40	16322	38036	43327	50838

Table 6.14: MAR stock selection: Number of trades

Year	Period, days	Intraday portfolio, stocks			
		1	3	5	10
2015	3	4359.91	1353.82	796.85	397.61
	5	3677.60	1172.65	701.50	363.74
	10	2972.14	982.10	617.17	348.35
	20	2479.20	862.87	572.45	345.11
	40	2191.47	780.10	553.90	349.91
2016	3	2754.67	996.70	633.09	352.90
	5	2316.12	888.66	584.29	341.51
	10	1941.77	786.20	547.56	341.19
	20	1731.07	740.10	537.81	351.00
	40	1623.78	721.29	545.56	368.19
2017	3	3279.04	1054.41	630.23	314.17
	5	2749.59	884.29	536.26	285.39
	10	2051.19	678.45	438.96	260.30
	20	1635.66	561.53	396.83	255.43
	40	1362.46	509.92	386.75	258.25

Table 6.15: MAR stock selection: Average profit per trade, USD

Year	Period, days	Intraday portfolio, stocks			
		1	3	5	10
2015	3	-1802.64	-519.31	-304.68	-157.38
	5	-1393.32	-417.62	-251.57	-138.33
	10	-958.23	-311.25	-203.26	-125.88
	20	-694.81	-251.78	-176.90	-121.68
	40	-584.17	-212.49	-167.43	-122.23
2016	3	-834.32	-308.29	-201.88	-122.16
	5	-625.71	-249.06	-172.05	-112.39
	10	-437.30	-198.55	-150.28	-107.55
	20	-338.14	-176.98	-142.57	-108.12
	40	-293.76	-171.40	-143.87	-111.39
2017	3	-1082.31	-351.63	-213.53	-113.26
	5	-858.34	-271.45	-168.76	-98.21
	10	-576.98	-187.81	-127.14	-85.48
	20	-411.78	-142.03	-109.32	-81.34
	40	-288.12	-121.45	-103.49	-80.39

Table 6.16: MAR stock selection: Average loss per trade, USD

Year	Period, days	Intraday portfolio, stocks			
		1	3	5	10
2015	3	4.26	2.59	1.17	-1.45
	5	11.68	6.71	3.87	-0.19
	10	21.08	12.02	7.25	1.57
	20	25.57	15.84	9.96	3.26
	40	**27.22**	17.73	11.66	4.77
2016	3	12.33	8.66	6.01	1.78
	5	28.30	16.65	10.89	4.30
	10	45.64	25.16	16.48	7.75
	20	49.08	29.22	19.94	10.51
	40	**49.60**	30.94	22.59	13.42
2017	3	2.97	1.80	0.92	-0.68
	5	6.06	3.52	1.95	-0.47
	10	11.47	6.23	3.46	0.35
	20	13.35	7.91	4.70	1.19
	40	**13.43**	8.99	5.71	2.06

Table 6.17: Moving Sharpe ratio stock selection: Profit, %

Year	Period, days	Intraday portfolio, stocks			
		1	3	5	10
2015	3	0.49	0.50	0.30	-0.47
	5	1.11	1.16	0.90	-0.09
	10	1.65	1.48	1.25	0.45
	20	2.25	1.87	1.57	0.81
	40	2.58	2.22	1.89	1.15
2016	3	1.03	1.08	0.92	0.40
	5	1.48	1.49	1.31	0.83
	10	2.19	1.91	1.70	1.27
	20	2.46	2.14	1.92	1.45
	40	2.50	2.26	2.04	1.63
2017	3	0.76	0.77	0.47	-0.51
	5	1.34	1.34	1.06	-0.34
	10	2.31	1.97	1.55	0.27
	20	3.33	2.40	1.97	0.80
	40	3.51	2.65	2.33	1.41

Table 6.18: Moving Sharpe ratio stock selection: Sharpe ratio

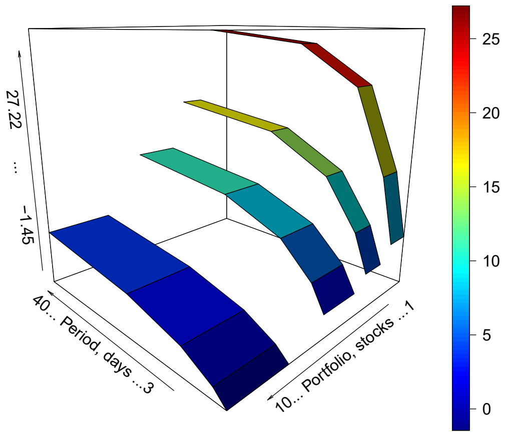


Figure 6.22: MS ratio stock selection (2015): Profit, %

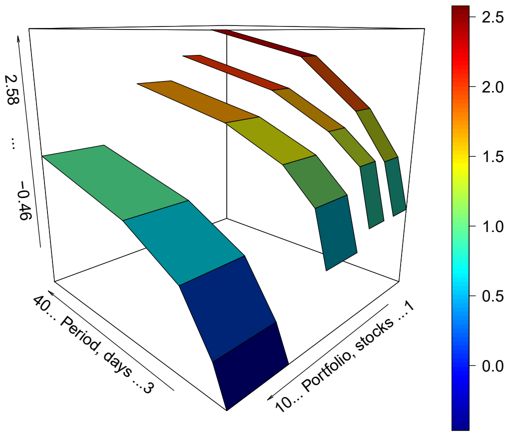


Figure 6.23: MS ratio stock selection (2015): Sharpe ratio

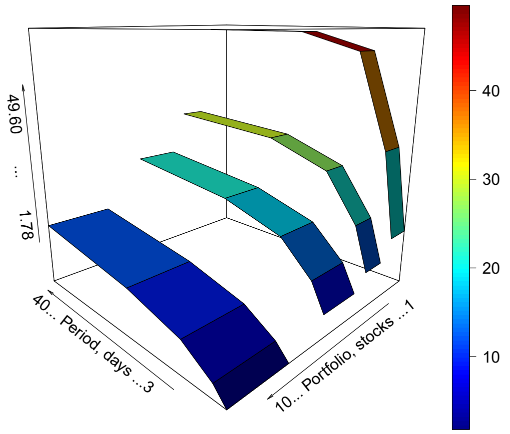


Figure 6.24: MS ratio stock selection (2016): Profit, %

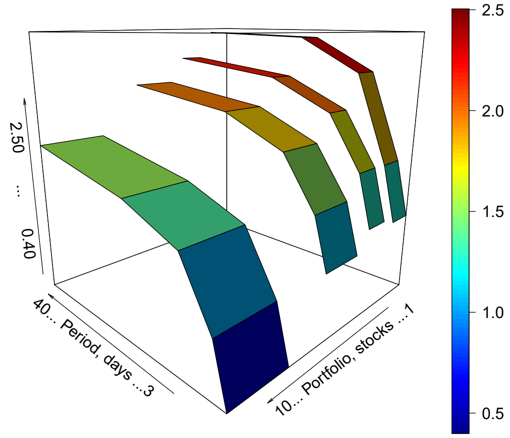


Figure 6.25: MS ratio stock selection (2016): Sharpe ratio

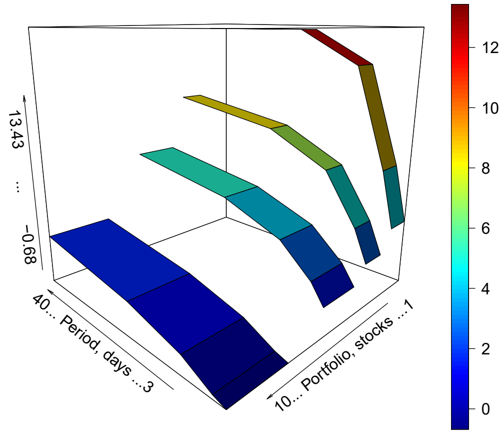


Figure 6.26: MS ratio stock selection (2017): Profit, %

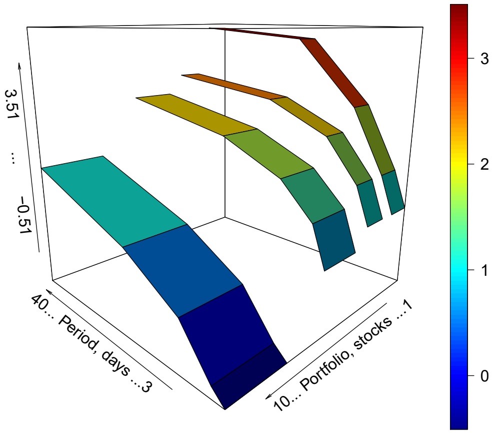


Figure 6.27: MS ratio stock selection (2017): Sharpe ratio

Year	Period, days	Intraday portfolio, stocks			
		1	3	5	10
2015	3	24.79	25.21	25.77	26.69
	5	22.95	24.11	25.00	26.32
	10	21.34	23.76	24.75	26.08
	20	20.38	23.66	24.60	26.06
	40	19.70	23.77	24.80	26.28
2016	3	24.28	24.70	25.04	25.76
	5	23.38	23.90	24.43	25.40
	10	22.50	23.77	24.34	25.44
	20	21.75	23.82	24.44	25.54
	40	21.51	24.07	24.64	25.69
2017	3	23.81	23.89	24.14	24.81
	5	22.15	22.67	23.16	24.12
	10	21.11	22.20	22.76	23.84
	20	20.55	22.28	22.83	23.93
	40	20.03	22.44	23.02	24.06

Table 6.19: Moving Sharpe ratio stock selection: Hit rate, %

Year	Period, days	Intraday portfolio, stocks			
		1	3	5	10
2015	3	7482	20181	30500	50920
	5	14384	31552	42854	61812
	10	25488	46304	56161	71727
	20	32581	55488	64507	78495
	40	35667	61049	70420	84953
2016	3	11234	28262	40996	61695
	5	23227	44667	55655	71388
	10	36489	59336	67088	78570
	20	40599	65923	72213	82584
	40	41615	69128	75374	86125
2017	3	4624	13024	19965	32697
	5	9248	21306	28668	39811
	10	16688	31483	37185	45351
	20	20783	36276	41044	48354
	40	22370	38702	43465	51102

Table 6.20: Moving Sharpe ratio stock selection: Number of trades

Year	Period, days	Intraday portfolio, stocks			
		1	3	5	10
2015	3	2728.04	953.96	598.26	324.03
	5	2105.66	807.68	530.68	308.56
	10	1720.45	725.27	507.72	314.23
	20	1594.78	713.52	516.56	327.75
	40	1551.42	703.11	520.87	338.36
2016	3	2220.52	800.89	513.49	297.82
	5	1795.48	718.72	489.89	303.01
	10	1650.64	706.65	504.73	324.57
	20	1605.37	711.42	526.68	348.38
	40	1581.20	708.66	545.96	372.92
2017	3	2124.15	711.06	443.63	246.99
	5	1493.31	565.55	380.44	231.16
	10	1166.23	491.79	355.19	231.75
	20	1065.57	482.71	362.14	241.65
	40	1006.61	487.81	371.86	250.81

Table 6.21: Moving Sharpe ratio stock selection: Average profit per trade, USD

Year	Period, days	Intraday portfolio, stocks			
		1	3	5	10
2015	3	-827.00	-304.52	-202.53	-121.85
	5	-522.26	-228.67	-164.94	-110.65
	10	-361.94	-192.04	-149.89	-107.93
	20	-309.68	-183.86	-148.03	-109.90
	40	-285.72	-181.10	-149.69	-112.94
2016	3	-567.83	-222.10	-151.97	-99.47
	5	-389.13	-176.78	-132.49	-95.09
	10	-317.81	-164.77	-129.87	-97.51
	20	-291.77	-164.20	-133.79	-102.38
	40	-281.51	-165.73	-138.74	-107.93
2017	3	-579.58	-205.22	-135.15	-84.28
	5	-341.20	-144.51	-105.85	-75.03
	10	-224.89	-114.91	-92.59	-71.52
	20	-194.76	-110.32	-92.31	-72.77
	40	-177.09	-111.18	-94.11	-74.14

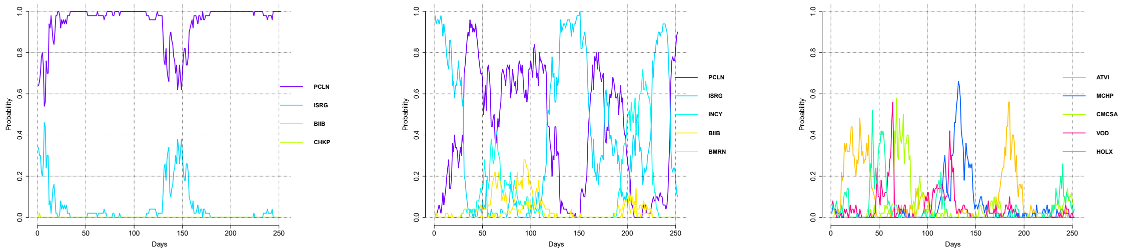
Table 6.22: Moving Sharpe ratio stock selection: Average loss per trade, USD

6.6 Probability of Stock Selection

This section gives insight into how often stocks were included in the intraday portfolio for the three ranking heuristics (MAR, MS, and Kelly). Table 6.23 shows the probability of each stock being selected for an intraday portfolio within each year for all portfolio sizes: 1, 3, 5 or 10 stocks (for clarity, only the top 10 most frequently occurring stock selections are reported). Figures 6.28 — 6.39 show the daily probability of the top five stocks (in 2015) as included in an intraday portfolio and, again, repeated for all portfolio sizes (1, 3, 5 or 10 stocks).

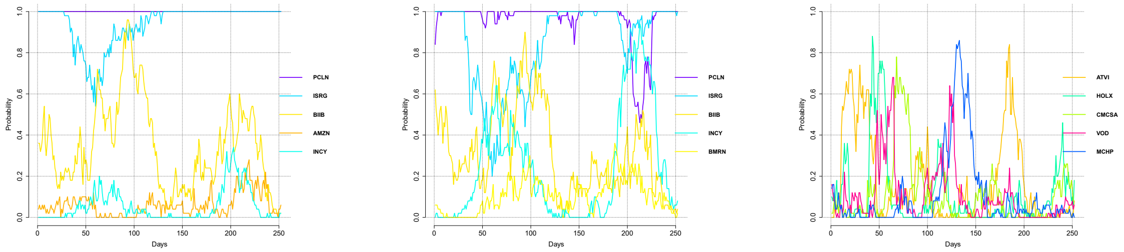
The strong preference for a very specific subset of stocks is again underlined. Indeed, the same stocks are always ranked 1, 2 and 3 for both the Moving Sharpe Ratio and Moving Average (Price) ranking heuristic. Increasing the size of each portfolio tends to result in the favored stocks being selected 100% of the time, with additional stocks then being added. Moreover, as the Intraday portfolio size increases, so does the likelihood of specific stocks being selected. Also of note is that there are orders of magnitude in the differences in the average stock price associated with the top ranked stocks, for example: PCLN or PIMCO Mutual Fund ($\approx$ \$5.50 per stock), ISRG or Intuitive Surgical Inc. ($\approx$ \$580 per stock), BIIB or Bilgen Inc. ($\approx$ \$235 per stock), AMZN or Amazon.com Inc. ($\approx$ \$1840 per stock), NTES or NetEase Inc. ($\approx$ \$270 per stock). This appears to imply that during intraday trading, the GP trading agent is attempting to maximize returns from the change in stock value.

The Kelly Criterion, on the other hand, appears to not represent an effective stock selection heuristic, with very little distinction appearing between the ranked stocks by the metric. Indeed, the stocks selected by Moving Sharpe Ratio and Moving Average (Price) were ranked very low by the Kelly Criterion (IDXX was the highest ranked at 39th). Conversely, both the Moving Sharpe Ratio and Moving Average (Price) ranking heuristics resulted in the same 7 stocks appearing in their respective top 10 selections, albeit not necessarily in the same order (Table 6.23), and with different intraday trading behaviors.



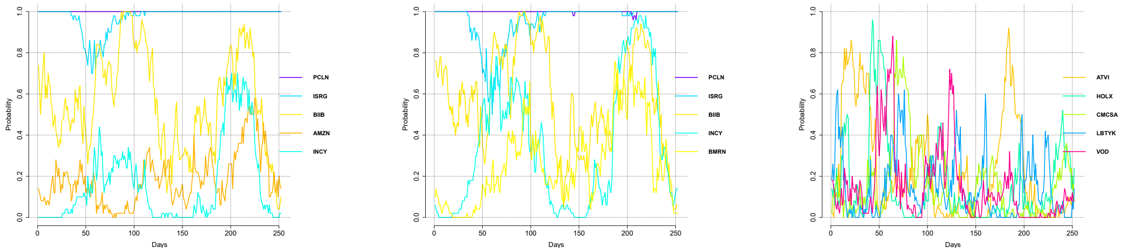
(a) MS (b) MAR (c) Kelly

Figure 6.28: Daily portfolio of 1 over 2015 (a) MS (b) MAR (c) Kelly.



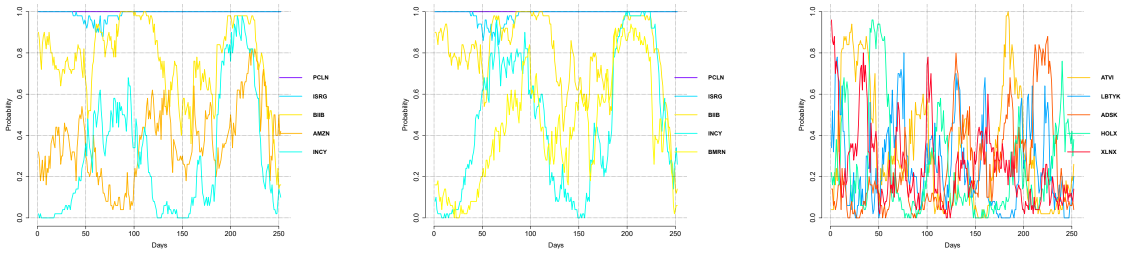
(a) MS (b) MAR (c) Kelly

Figure 6.29: Daily portfolio of 3 over 2015 (a) MS (b) MAR (c) Kelly.



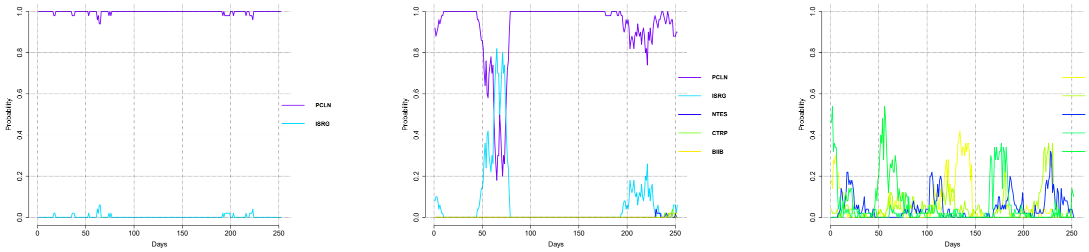
(a) MS (b) MAR (c) Kelly

Figure 6.30: Daily portfolio of 5 over 2015 (a) MS (b) MAR (c) Kelly.



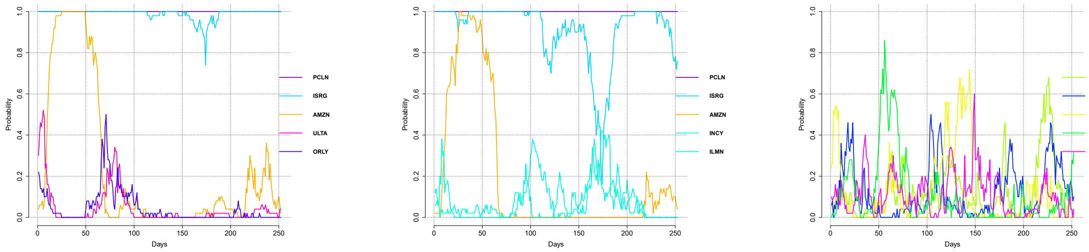
(a) MS (b) MAR (c) Kelly

Figure 6.31: Daily portfolio of 10 over 2015 (a) MS (b) MAR (c) Kelly.



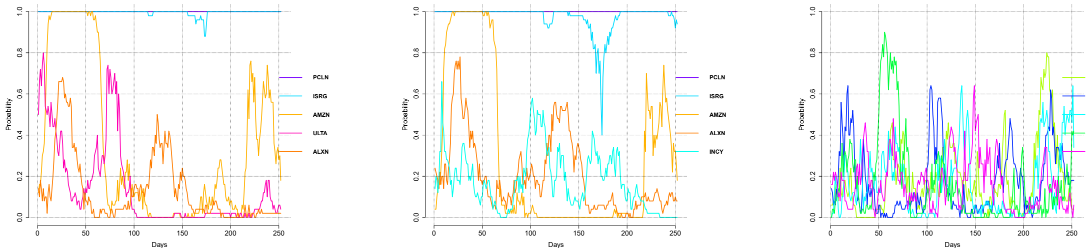
(a) MS (b) MAR (c) Kelly

Figure 6.32: Daily portfolio of 1 over 2016 (a) MS (b) MAR (c) Kelly.



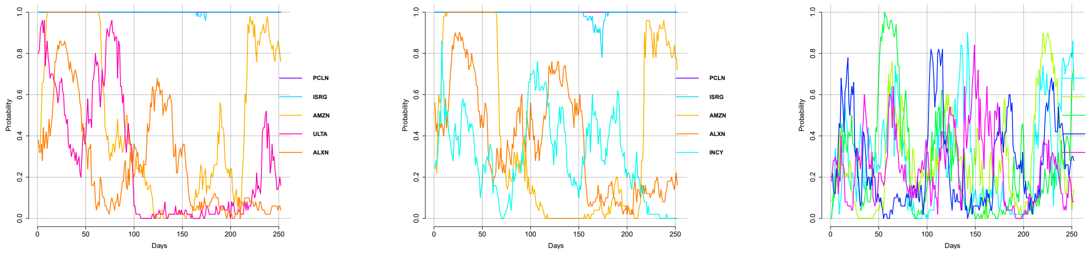
(a) MS (b) MAR (c) Kelly

Figure 6.33: Daily portfolio of 3 over 2016 (a) MS (b) MAR (c) Kelly.



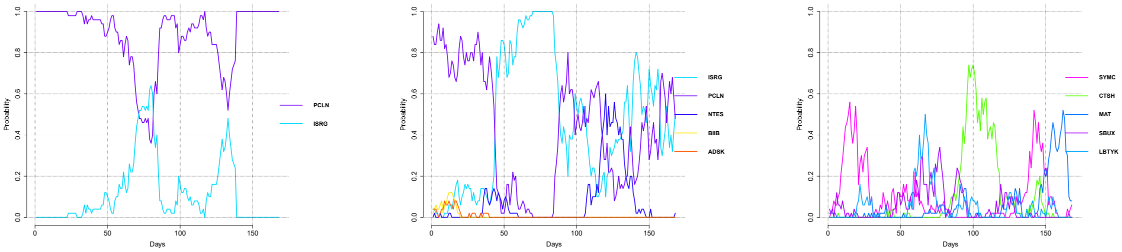
(a) MS (b) MAR (c) Kelly

Figure 6.34: Daily portfolio of 5 over 2016 (a) MS (b) MAR (c) Kelly.



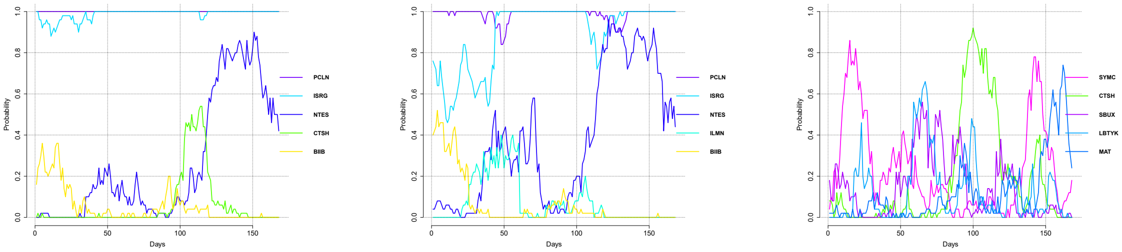
(a) MS (b) MAR (c) Kelly

Figure 6.35: Daily portfolio of 10 over 2016 (a) MS (b) MAR (c) Kelly.



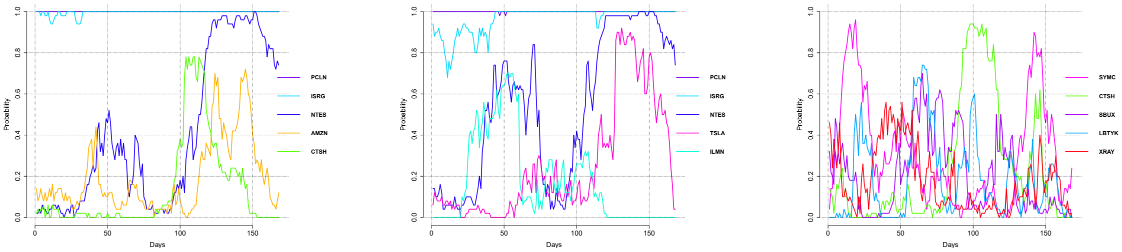
(a) MS (b) MAR (c) Kelly

Figure 6.36: Daily portfolio of 1 over 2017 (a) MS (b) MAR (c) Kelly.



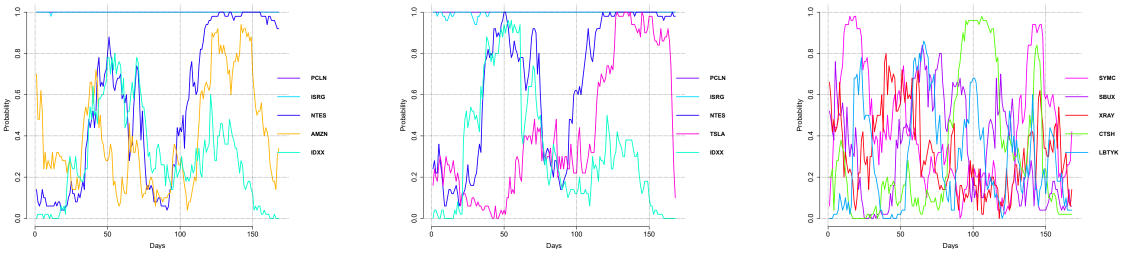
(a) MS (b) MAR (c) Kelly

Figure 6.37: Daily portfolio of 3 over 2017 (a) MS (b) MAR (c) Kelly.



(a) MS (b) MAR (c) Kelly

Figure 6.38: Daily portfolio of 5 over 2017 (a) MS (b) MAR (c) Kelly.



(a) MS (b) MAR (c) Kelly

Figure 6.39: Daily portfolio of 10 over 2017 (a) MS (b) MAR (c) Kelly.

	Year												Average prob.
	2015				2016				2017				
	Intraday portfolio size, stocks.												
	1	3	5	10	1	3	5	10	1	3	5	10	
Selection using Moving Sharpe Ratio (MS)													
PCLN	.948	1	1	1	.998	1	1	1	.885	1	1	1	.986
ISRG	.052	.942	.973	.992	.002	.989	.996	.999	.115	.988	.996	1	.754
BIIB	.000	.349	.568	.772	0	.021	.066	.191	0	.055	.128	.257	.201
AMZN	0	.06	.183	.371	0	.243	.351	.491	0	.052	.192	.397	.195
NTES	0	.012	.042	.104	0	.018	.053	.153	0	.268	.406	.548	.134
ORLY	0	.046	.116	.225	0	.043	.129	.251	0	.026	.084	.224	.095
IDXX	0	.034	.096	.241	0	.018	.078	.199	0	.034	.111	.298	.093
ILMN	0	.025	.093	.231	0	.027	.067	.174	0	.035	.091	.211	.08
TSLA	0	.012	.046	.148	0	.02	.079	.198	0	.018	.119	.289	.077
ULTA	0	.018	.043	.118	0	.045	.141	.281	0	.011	.055	.195	.076
Selection using Moving Average of Daily Returns (MAR)													
PCLN	.417	.957	.999	1	.924	1	1	1	.424	.987	1	1	.892
ISRG	.402	.852	.952	.988	.072	.913	.974	.994	0	.894	.965	.997	.75
BIIB	.042	.314	.587	.803	.001	.017	.056	.194	.007	.066	.133	.27	.207
NTES	.003	.038	.089	.192	.001	.049	.114	.237	0	.381	.536	.665	.192
INCY	.11	.242	.347	.475	0	.083	.178	.305	0	.017	.055	.138	.163
ALXN	.003	.076	.18	.358	0	.057	.204	.369	.005	.03	.075	.174	.128
TSLA	.001	.027	.074	.2	0	.044	.144	.29	.003	.051	.255	.421	.126
ILMN	0	.027	.107	.298	0	.06	.128	.258	.006	.075	.171	.313	.12
AMZN	0	.013	.064	.271	0	.195	.295	.419	0	.001	.019	.154	.119
BMRN	.014	.105	.264	.438	0	.018	.054	.118	.002	.008	.053	.144	.102
Selection using Kelly Criterion (Kelly)													
SYMC	.021	.062	.099	.171	.035	.108	.165	.259	.104	.234	.309	.421	.166
CMCSA	.048	.109	.154	.238	.049	.123	.181	.285	.034	.093	.145	.261	.143
LBTYK	.031	.098	.152	.254	.024	.065	.1	.181	.05	.128	.19	.289	.13
MAT	.036	.101	.142	.221	.015	.05	.086	.167	.059	.121	.175	.268	.12
DISCK	.018	.064	.103	.173	.032	.085	.139	.228	.047	.106	.148	.23	.114
CA	.035	.082	.123	.2	.055	.12	.164	.247	.013	.051	.093	.175	.113
XRAY	.024	.068	.105	.187	.014	.047	.079	.157	.039	.114	.182	.319	.111
SBUX	.003	.02	.038	.096	.027	.08	.123	.209	.052	.142	.209	.333	.111
FOX	.023	.066	.098	.171	.042	.094	.127	.184	.044	.103	.144	.218	.109
VOD	.046	.107	.151	.239	.033	.076	.113	.178	.023	.061	.092	.146	.105

Table 6.23: Probability of stock selection for 10 highest ranked stocks as defined across average across all portfolios.

6.7 Fixed Set of TI

In order to evaluate the dependence/independence of the proposed intraday stock selection algorithms from the trading signal generating algorithm, the second set of trading signals was generated with a benchmarking framework that evolves DT population against a fixed set of the popular technical indicators and trading rules as described in Section 3.4. The average results over 50 simulation runs with a fixed set of TIs for two proposed intraday stock selection algorithms (MAR and MS), and Kelly Criterion is shown in Tables 6.24, 6.25 and 6.26 respectively.

Year	Period days	Profit, % Intraday portfolio, stocks				Sharpe ratio Intraday portfolio, stocks			
		1	3	5	10	1	3	5	10
2015	3	-2.45	-2.40	-2.47	-2.74	-0.29	-0.49	-0.64	-0.98
	5	-2.30	-2.18	-2.27	-2.58	-0.30	-0.47	-0.61	-0.90
	10	-0.96	-1.51	-1.79	-2.22	-0.14	-0.35	-0.54	-0.91
	20	**0**	-2.07	-2.02	-2.40	0	-0.47	-0.59	-0.95
	40	-2.21	-2.17	-2.22	-2.55	-0.32	-0.53	-0.68	-1.06
2016	3	**-0.02**	-0.93	-1.05	-1.56	-0.02	-1.18	-0.24	-0.48
	5	-0.04	-1.05	-1.23	-1.66	-0.01	-0.22	-0.31	-0.56
	10	-0.69	-1.25	-1.48	-1.81	-0.10	-0.27	-0.40	-0.66
	20	-0.52	-0.95	-1.33	-1.63	-0.08	-0.20	-0.35	-0.59
	40	-1.89	-1.78	-1.77	-2.00	-0.25	-0.44	-0.55	-0.85
2017	3	-0.89	-1.22	-1.34	-1.51	-0.26	-0.52	-0.73	-1.14
	5	-1.27	-1.28	-1.33	-1.46	-0.37	-0.61	-0.87	-1.27
	10	**-0.67**	-1.01	-1.01	-1.20	-0.18	-0.52	-0.67	-1.05
	20	-1.71	-1.45	-1.39	-1.40	-0.54	-0.78	-0.94	-1.29
	40	-1.37	-1.10	-1.12	-1.28	-0.43	-0.65	-0.84	-1.31

Table 6.24: Fixed set of TIs. MAR stock selection: Profit and Sharpe ratio

Year	Period days	Profit, % Intraday portfolio, stocks				Sharpe ratio Intraday portfolio, stocks			
		1	3	5	10	1	3	5	10
2015	3	-1.69	-1.86	-1.93	-2.34	-0.27	-0.49	-0.60	-0.91
	5	-1.89	-1.59	-1.78	-2.24	-0.33	-0.44	-0.60	-0.96
	10	-0.80	-0.89	-1.32	-2.01	-0.14	-0.27	-0.50	-1.02
	20	-0.25	-1.25	-1.64	-2.14	-0.06	-0.41	-0.66	-1.21
	40	**0.61**	-0.80	-1.28	-1.93	0.18	-0.28	-0.57	-1.08
2016	3	-0.90	-1.06	-1.15	-1.41	-1.14	-0.31	-0.39	-0.56
	5	-0.24	-0.50	-0.84	-1.45	-0.05	-0.16	-0.31	-0.61
	10	0.845	-0.33	-0.72	-1.37	0.14	-0.11	-0.28	-0.67
	20	1.39	0.04	-0.47	-1.32	0.26	0.01	-0.17	-0.63
	40	**2.47**	0.32	-0.35	-1.21	0.61	0.11	-0.15	-0.66
2017	3	-0.48	-0.77	-0.94	-1.23	-0.17	-0.39	-0.64	-0.97
	5	-0.71	-0.78	-0.91	-1.25	-0.23	-0.38	-0.54	-1.04
	10	-0.95	-0.93	-0.95	-1.17	-0.34	-0.56	-0.71	-1.14
	20	-0.42	-0.78	-0.99	-1.23	-0.18	-0.61	-0.91	-1.44
	40	**0.63**	-0.28	-0.59	-0.98	0.55	-0.24	-0.60	-1.25

Table 6.25: Fixed set of TIs. MS stock selection: Profit and Sharpe ratio

Year	Period days	Profit, % Intraday portfolio, stocks				Sharpe ratio Intraday portfolio, stocks			
		1	3	5	10	1	3	5	10
2015	3	-1.64	-1.98	**-1.06**	-2.39	-0.28	-0.55	-0.70	-1.02
	5	-2.04	-1.91	-1.95	-2.19	-0.33	-0.53	-0.65	-0.91
	10	-2.05	-1.88	-2.04	-2.28	-0.37	-0.55	-0.70	-1.00
	20	-2.79	-2.37	-2.43	-2.58	-0.55	-0.71	-0.92	-1.34
	40	-2.55	-2.50	-2.46	-2.57	-0.48	-0.79	-0.93	-1.29
2016	3	-1.47	**-1.02**	-1.06	-1.43	-0.23	-0.30	-0.38	-0.60
	5	-1.35	-1.53	-1.70	-1.80	-0.29	-0.47	-0.61	-0.80
	10	-1.66	-1.50	-1.59	-1.77	-0.34	-0.50	-0.60	-0.82
	20	-1.27	-1.66	-1.73	-1.85	-0.24	-0.51	-0.64	-0.88
	40	-1.73	-1.72	-1.63	-1.85	-0.36	-0.55	-0.64	-0.95
2017	3	**-0.63**	-0.65	-0.92	-1.21	-0.19	-0.28	-0.56	-0.98
	5	-1.04	-0.95	-0.99	-1.12	-0.37	-0.57	-0.69	-1.05
	10	-0.75	-0.89	-0.95	-1.12	-0.28	-0.52	-0.71	-1.07
	20	-0.66	-0.88	-0.90	-1.05	-0.27	-0.56	-0.70	-1.03
	40	-0.69	-0.75	-0.78	-1.00	-0.25	-0.51	-0.70	-1.16

Table 6.26: Fixed set of TIs. Kelly stock selection: Profit and Sharpe ratio

Based on the obtained results (Tables 6.24, 6.25 and 6.26) one can conclude that the results of daily stock selection in the case of two proposed algorithms do not depend on the way of generating of trading signals. In addition, again it was confirmed that coevolving both TI and DT together significantly outperforms evolving the DT based on predefined TI (as in [68]).

6.8 Discussion

A clear preference is exhibited for adopting a ranking based on either a simple Moving Average, or the Moving Sharpe Ratio (with Moving Sharpe Ratio showing better results over a simple Moving Average of Daily Returns in all cases), both in terms of the profitability and the Sharpe Ratio. Moreover, both schemes perform significantly better than the Kelly Criterion (a popular method for long-term investment [135, 106, 26, 87]).

The Moving Sharpe Ratio outperforms other investigated ways of prioritizing specific stocks for frequent intraday trading using the proposed FXGP algorithm. Figures 6.40 and 6.41 summarize the percent Profit and Sharpe Ratio per year over all 50 trials for the preferred parameterization (40 day estimation period for the ranking statistic, 1 stock selected per intraday trading period). Every run is $> 10\%$ profitable and returns Sharpe Ratios with a minimum of 2. In comparison, adopting a buy-and-hold investment strategy for all 86 stocks in the portfolio[7] results in the investment outcomes of Table 6.1.

In short, even though the frequent intraday trading scenario incurs a transaction cost per trade, and there are typically tens of thousands of trades (Table 6.20), only 2 of 20 parameterizations perform worse than the buy-and-hold strategy in 2015 and all perform better than buy-and-hold in 2016. Only 5 of 20 parameterizations perform worse than the buy-and-hold strategy in the 8 months of 2017 (Table 6.17).

There are no parameterizations for which a stock ranking performed using the Kelly Criterion approaches that of either the Moving Average or the Moving Sharpe Ratio. Performing a t-test between the average % Profit (Sharpe Ratio) of the 50 simulations in Tables 6.5, 6.11 and 6.17 (Tables 6.6, 6.12 and 6.18) confirms that in all cases the Moving Sharpe Ratio performs significantly better than ranking using

---

[7] All stocks from the portfolio are held, as there is no basis for selecting a subset of specific stock.

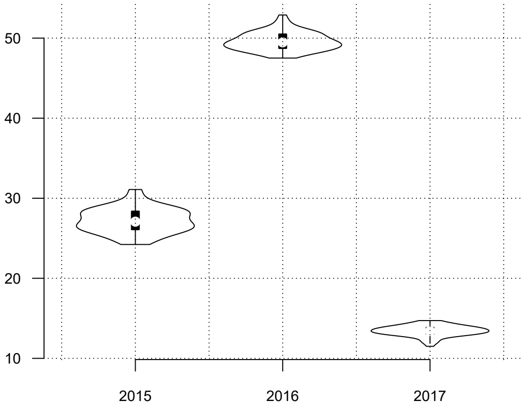


Figure 6.40: Moving Sharpe ratio algorithm. Profits (%) distribution over 50 simulations

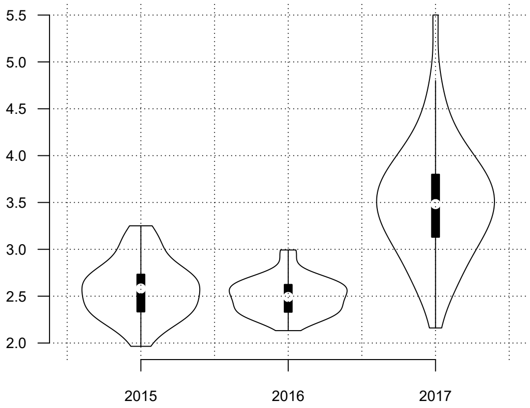


Figure 6.41: Moving Sharpe ratio algorithm. Sharpe ratios distribution over 50 simulations

a Moving Average or Kelly Criterion, Table 6.27. To check the distribution of the results the Shapiro-Wilk normality test was used. The p-values of all test are $\geqslant 0.05$ that allows us to assume the normality of all distributions (Table 6.28).

Test pair	% Profit	Sharpe Ratio
MS versus MAR	5.1×10^{-8}	3.5×10^{-13}
MS versus Kelly	3.9×10^{-11}	1.3×10^{-24}

Table 6.27: Pairwise t-test for significant differences between average Profit (Sharpe Ratio) over 50 parameterizations of ranking metrics. Assuming a 99% confidence interval and using the Bonferroni-Dunn Post Hoc Test, then $\alpha = \frac{0.01}{3} = 0.0033$. All the p-values are smaller than α, implying that the pairwise differences are all significant.

Metric	2015	2016	2017
MS	0.36	0.19	0.28
MAR	0.15	0.08	0.05
Kelly	0.6	0.8	0.28

Table 6.28: Shapiro-Wilk normality test, p-values

In addition, a general bias towards trading with single stocks per day was evident. That is to say, although increasing the number of stocks traded per day resulted in a lower 'average loss per trade' the corresponding 'average profit per trade' was also much lower. Thus, the resulting Sharpe Ratio (a measure of average profit-to-risk) was actually preferable for the case of performing frequent intraday trading with one stock per day.[8] It is a mark of the accuracy of the estimated returns from the GP trading agents that such specific recommendations could be made without having a detrimental effect on the performance of the investment strategy, i.e. risk.

---

[8] Naturally, the recommended stock is free to change each day.

Chapter 7

On the Effect of Hidden Trading Costs

Section 2.6 reviewed the significance of transactions costs in automated trading agent design. The specific interest here is to identify to what degree (if any) an automated trading system utilizing historical rates to construct trading rules for real-time intraday trading may benefit from fixed or floating spreads. This is potentially an important decision because different brokerages may differ in terms of whether the bid-ask spread is fixed or floating. Moreover, if there is a difference, how large or small might the size of a fixed versus floating spread have to be before comparable results appear? This chapter investigates the impact of the bid-ask spreads, a form of hidden cost, on the results of backtesting. Backtesting is the general method for seeing how well a strategy or model would have done using historical data. If backtesting works, traders and analysts may have the confidence to employ it going forward [69]. It concentrates on the nature (fixed or floating) of bid-ask spreads (hereafter 'spread') and investigates the different impact of two types of spread on the effectiveness of an automated trading system [96]. Four fixed spreads (one, two, five and ten pips) and a floating spread with a median value of two pips are investigated.

7.1 Experimental Framework

Figure 7.1 shows the experimental framework that was used to evaluate the influence of bid-ask spreads on the effectiveness of the automated trading system. Before each trading session begins the proposed FXGP algorithm is used to simulate trading of each stock over 40 preceding trading days using historical rates in the form of 1-minute candlesticks. The results of the simulation are used to rank each stock from the portfolio utilizing the best stock selection algorithm — the Moving Sharpe ratio (Section 6.3), and the stock with the highest K rank are selected for trading at the next trading day. During trading day t, K stock is selected to make investments and traded with a money management policy.

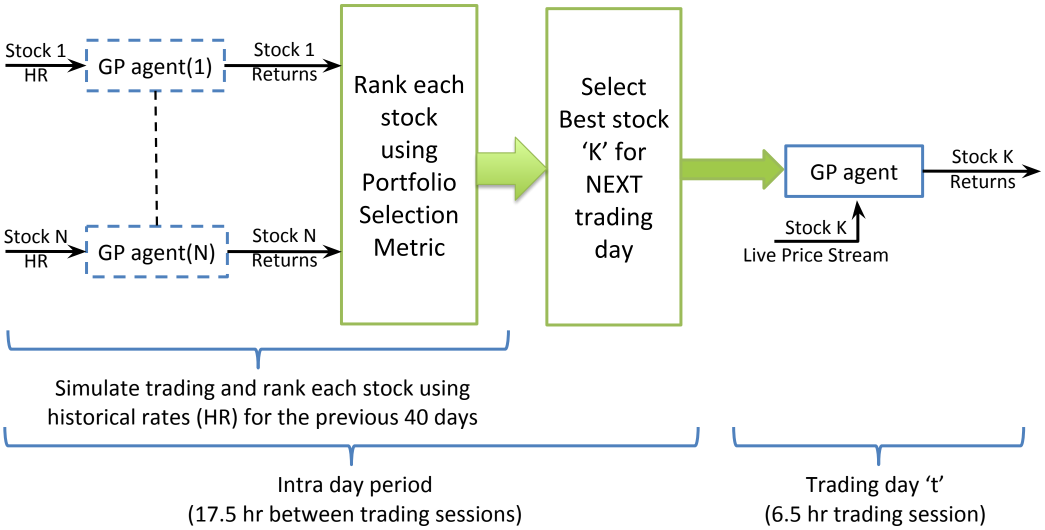


Figure 7.1: Experimental framework. Dashed boxes represent agents estimating a return on stock 'i' over preceding 40 trading days using historical rates. Solid box indicates the agent that performs intraday trading with the selected best stock.

7.2 Source Data

For continuity with Chapter 6, the source stock data will again take the form of the NASDAQ 100. Historical rates ('Bid', 'Mid' and 'Ask' prices) were obtained through an Interactive Brokers Group, Inc. (demo account) for the period from August 1, 2014, to August 31, 2017. Adopting data from the NASDAQ is expected to be representative of a market with 'high' levels of volatility ([73]). All stocks with missing or out of range date and time stamps and stocks with missing, '0' or negative prices and stocks with negative spreads were excluded. The resulting set includes following 77 stocks: AAL, AAPL, ADBE, ADI, ADP, AKAM, ALXN, AMAT, AMGN, AMZN, ATVI, BIDU, BIIB, CA, CELG, CERN, CHKP, CMCSA, CSCO, CSX, CTAS, CTRP, CTSH, CTXS, DISCK, DISH, EA, EBAY, ESRX, FAST, FB, FOX, FOXA, GILD, HAS, HOLX, ILMN, INCY, INTC, INTU, ISRG, KLAC, LBTYK, LRCX, MAR, MAT, MCHP, MDLZ, MSFT, MU, MXIM, NFLX, NVDA, ORLY, PAYX, PCAR, PCLN, QCOM, ROST, SBUX, SHPG, SIRI, STX, SWKS, SYMC, TSCO, TSLA, TXN, ULTA, VIAB, VOD, VRSK, VRTX, WBA, WDC, WYNN, XLNX.

7.3 Trading Conditions and Experimental Methodology

All experiments were performed assuming the following common trading conditions: Initial account balance \$10 000 000, flat rate \$5 per trade and only 10% of the account balance at the beginning of each day is used to trade stocks. This work investigates six different bid-ask spreads: 0 pips (0.0 USD), 1 pip (0.01 USD), 2 pips (0.02 USD), 5 pips (0.05 USD), 10 pips (0.1 USD) and floating spread that is defined as a difference in bid and ask prices of a stock. The distribution of floating spreads of all 77 stocks over 2015-2017 is shown in the Table 7.1 and Figure 7.2.

	min	1st quartile	median	3rd quartile	max
Spread, USD	0	0.01	0.02	0.05	141.69

Table 7.1: Floating spreads distribution 2015-2017

Bid and Ask prices with specific spreads were established by subtracting (for 'Bid' prices) or adding (for 'Ask' prices) a half of the spread size from/to the corresponding 'Mid' price (Section 7.2). The FXGP algorithm (Section 7.1) was used to simulate trading activity and performed 50[1] sets of simulation runs for each spread value, where each set includes trading signals for all 77 stocks and covers 2015–2017 years.

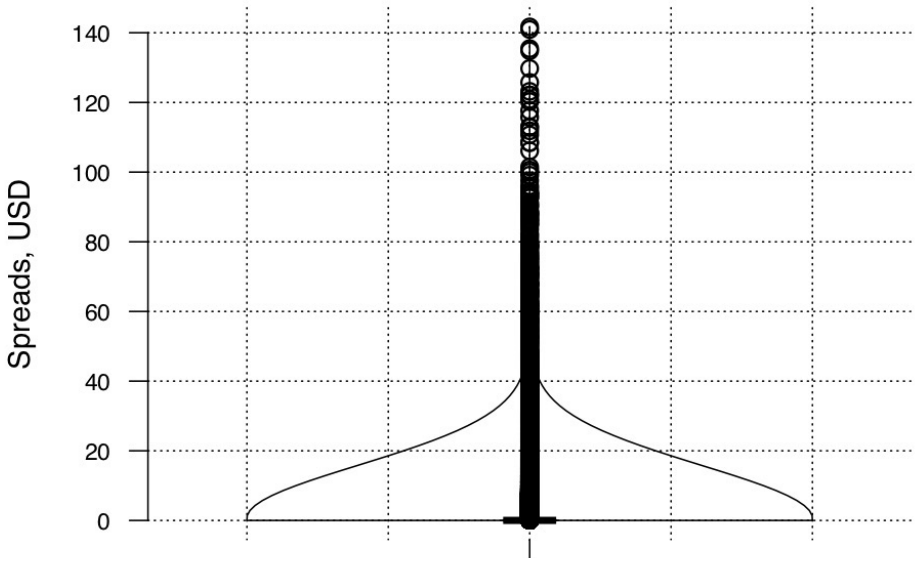
In addition, a baseline investment strategy using 'buy-and-hold' with the same spreads and trading conditions: 10% of the account balance to buy stocks and distribute this evenly amongst all 77 stocks will be established. Previous research has shown that such a 'naive' approach to portfolio management can be more effective than optimization methods, especially when transaction costs are included ([83, 38]). The profit of a buy-and-hold strategy was calculated annually (2015, 2016 and 2017) and for the whole period (2015-2017).

To evaluate the performance the following metrics were calculated:

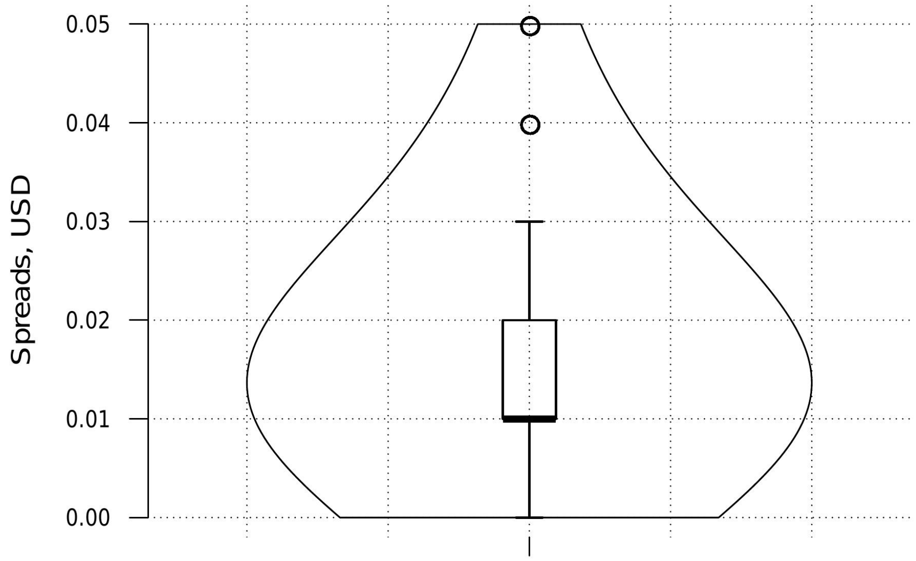
- Profit (%) — is the difference between an opening account balance and a balance at the end of the period including any transaction costs.
- Sharpe ratio (monthly returns) — defined using Equation (6.2) with R_{mean} and $R_{deviation}$ estimated monthly over the *entire* year (12 months).

---

[1] Due to a large number of stocks and one-minute candlesticks over three years, we reduced the number of simulation runs from 100 to 50.



(a) Full range



(b) 0-0.05 USD

Figure 7.2: Floating spreads distribution 2015-2017 (a) Full range (b) 0-0.05 USD.

- Hit rate (%) — attempts to express the accuracy of buy/sell signals relative to the outcome of the following candlestick. Thus, if the GP trading agent issued a buy (sell) signal and the price increased (decreased) at the next candlestick, this would be considered a 'hit'.

- Number of trades — is the raw count of the number of trades placed by the GP trading agents in total.

- Average profit per trade (USD) — calculated as a sum of all trades within a year with a positive value divided by the number of such trades.

- Average loss per trade (USD) — calculated as a sum of all trades within a year with a negative value divided by the number of such trades.

- 'Average profit'/'Average loss' ratio — calculated as 'Average profit per trade' divided by 'Average loss per trade.'

7.4 Results

7.4.1 Buy-and-Hold

A buy-and-hold strategy was used to provide a comparative baseline that is widely used in practice. As emphasized in Section 6.5.1, buy-and-hold is most competitive when market volatility is less [20]. From the perspective of the data appearing in this study, this occurs in 2016 and 2017. Indeed, current benchmarking practice, for example with deep learning solutions, rely on buy-and-hold baselines [12]. In the specific context of this chapter, a buy-and-hold strategy minimizes costs because the number of trades is minimized. Thus, any adaptive scheme for intraday trading has to be able to offset the cost of potentially performing many trades per day, by achieving much higher returns. Given that the time period of a price bar is 1-minute, there are more sources of noise that may hide trends in the data, as well as exasperate the likelihood of seeing variable spreads.

Profit of the 'buy-and-hold' strategy is defined by assuming that one buys the stocks with the 'Open' price of the first 1-minute candlestick of the evaluated period and sells them with the 'Open' price of the last candlestick of the same period. The process is repeated for all five fixed spreads and for the floating spread, with the results summarized in Table 7.2.

In the case of 'buy-and-hold' strategy, commission is paid only twice per trading period: annually or once per three years, i.e. two scenarios are considered, the profit of a buy-and-hold strategy was calculated annually (2015, 2016 and 2017) or for the whole period (2015-2017). Thus, from Table 7.2 it is apparent that the fixed spreads barely affect the results of the 'buy-and-hold' strategy. Some degradation of the profits is seen in the case of floating spreads, and can be explained as follows: the beginning and the end of a year represent one of the most unpredictable times to perform trading. As a consequence, the floating spreads at this time can be much higher, in some cases reaching more than a hundred dollars (Table 7.1 and Figure 7.2). Also evident is the increasing returns from 2015 through 2016 and 2017. This follows from the reduced market volatility as the years progress,[2] further favoring the buy-and-hold strategy.

Metric	Spread, USD					
	0	0.01	0.02	0.05	0.1	floating
Profit, 2015 (%)	0.664	0.662	0.66	0.653	0.642	0.474
Profit, 2016 (%)	1.266	1.264	1.262	1.255	1.244	1.19
Profit, 2017 (%)	1.732	1.730	1.728	1.723	1.713	1.618
Profit, 2015-2017 (%)	4.279	4.277	4.275	4.268	4.257	4.051

Table 7.2: Profit, buy-and-hold strategy

7.4.2 Automated Intraday Trading

This section presents the results obtained with the proposed FXGP framework (Section 7.1) using rates with a floating and the five fixed spreads. Results are averaged over 50 simulations (runs) for each spread value, and presented in Tables 7.3, 7.4 and 7.5. Figures 7.3(a), 7.5(a) and 7.7(a) show the annual distribution of profits and Figures 7.3(b), 7.5(b) and 7.7(b) show the distribution of the Sharpe ratios as averaged over all 50 simulations for each year.

Unlike the 'buy-and-hold' strategy, the results of the automated intraday trading system are strongly dependent on the nature (fixed or floating) and the size of the

---

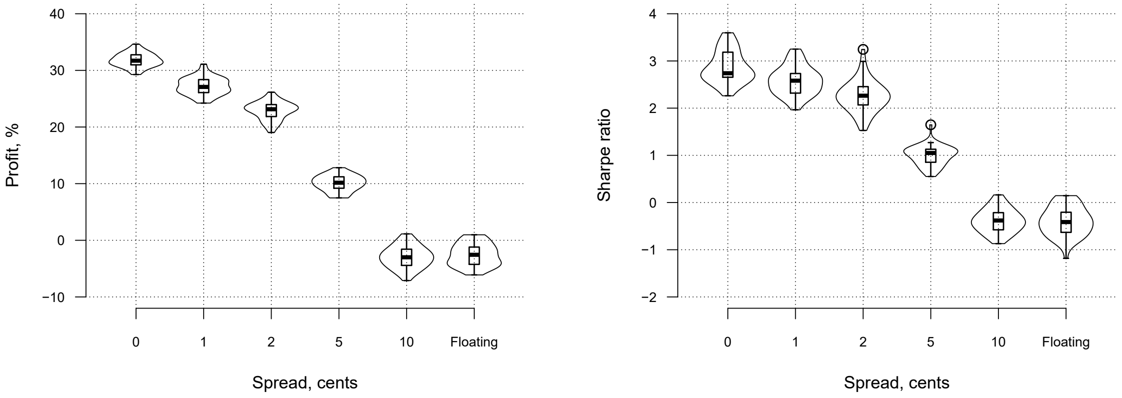
[2] See for example the decreasing Skewness and Kurtosis of NASDAQ 100 stocks, Appendix A.

bid-ask spread. Annual profits and Sharpe ratios drop with increasing spread and, at the same time, the intraday trading algorithm tends to trade less frequently and, therefore, to reduce the size of the commission paid, i.e. the automation identified the frequency of the trading necessary to provide this optimization.

The worst results appeared in the case of floating spreads. Even with the fixed bid-ask spread 0.1 USD, two years were profitable (2016 and 2017) while with the floating spread all three years were unprofitable. Taking into consideration that the median value of the floating spread is as low as 0.02 USD and that 75% of all floating spread values are less than 0.05 USD (Table 7.1), it is save conclude that the nature of the bid-ask spread is even more important than its magnitude. Specifically, the floating spread is capable of returning a higher per trade profit (hit rate is twice that of the fixed spreads) and average profit (per trade) is also always significantly higher, but this is outweighed by the much higher average loss (per trade). Given the significantly lower number of trades in each period, it is apparent that the floating spread results in a trading policy being adopted in which a relatively small number of trades are placed, but of high value. Conversely, all the fixed spread scenarios result in trading policies being adopted in which many more smaller value trades are placed. Such a strategy results in a better average profit to loss ratio being maintained (typically > 4), thus an annual Sharpe Ratio above unity. This level of trading performance is never achieved under the variable spread trading environment. Finally, in order to make more apparent the interaction between different performance metrics and the forms of spread, all the annual metrics were normalized (Tables 7.3, 7.4 and 7.5) and plotted: Figures 7.4, 7.6 and 7.8.

Metric	Spread, USD					
	0	0.01	0.02	0.05	0.1	floating
Profit (%)	31.82	27.22	22.88	10.09	-2.97	-2.6
Sharpe ratio	2.87	2.58	2.25	0.99	-0.386	-0.4
Hit rate (%)	19.78	19.7	19.72	20.74	23.18	42.45
Num. of trades	36727	35667	34312	24702	10883	427
Average Profit (USD)	1561	1551	1537	1550	1746	8471
Average Loss (USD)	277	286	295	355	567	7299
$\frac{\text{Avg. Profit}}{\text{Avg. Loss}}$	5.64	5.43	5.22	4.37	3.08	1.16

Table 7.3: Results (averages of 50 runs) of trading simulation over 2015. Sharpe Ratio is estimated as a monthly return. Both Avg. profit and Avg. loss are estimated per trade.



(a) Profits

(b) Sharpe

Figure 7.3: Distribution over 50 runs of (a) Profit and (b) Sharpe ratios in 2015

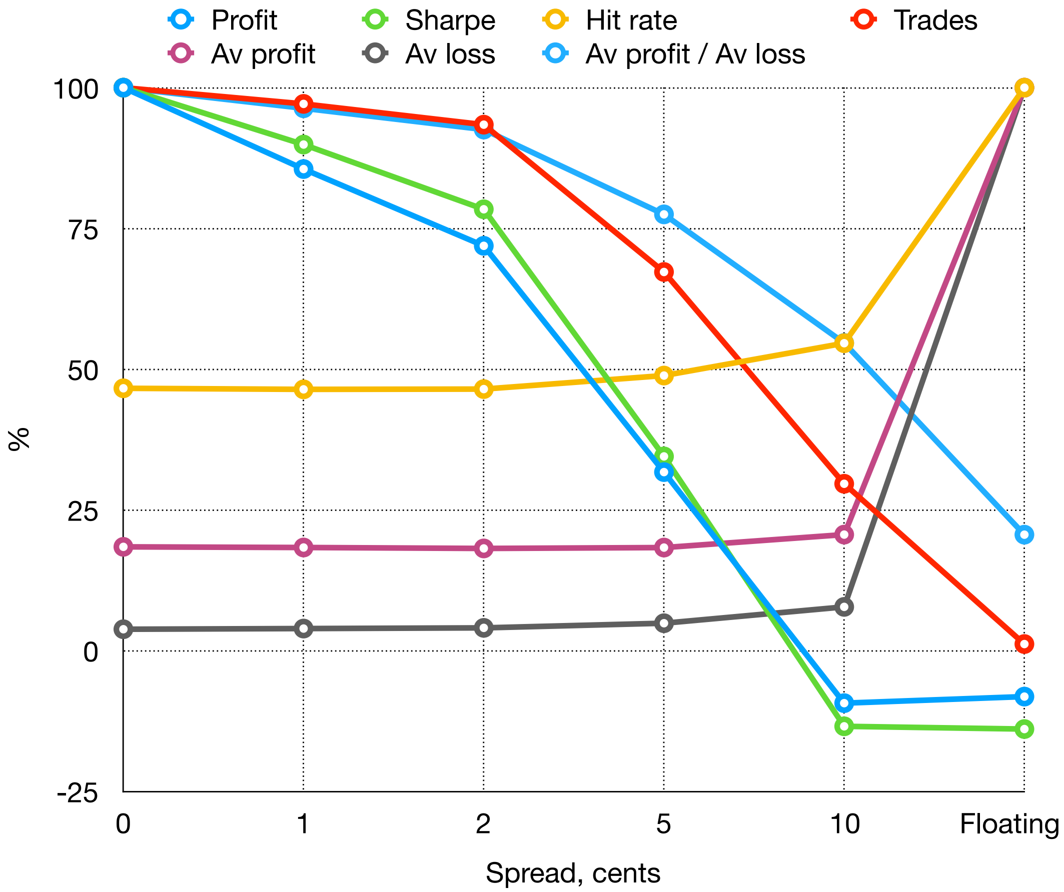
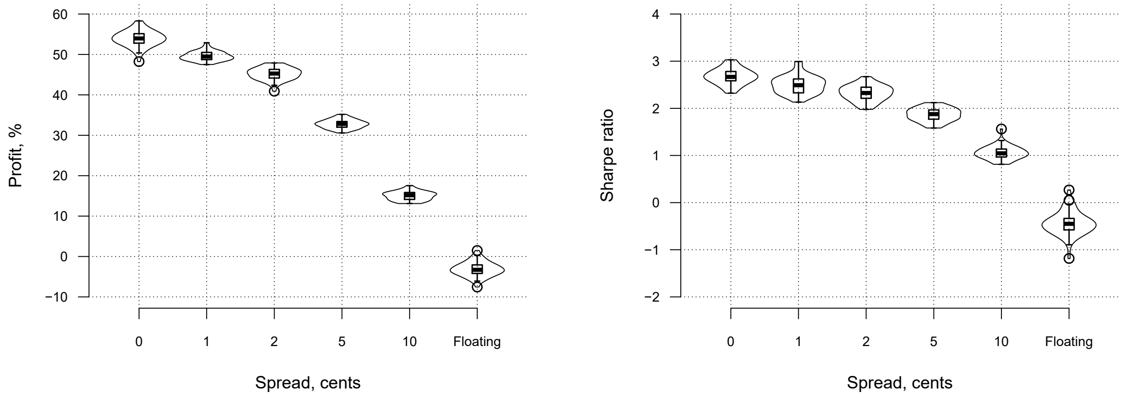


Figure 7.4: Normalized metrics 2015

Metric	Spread, USD					
	0	0.01	0.02	0.05	0.1	floating
Profit (%)	53.96	49.6	45.25	32.8	15	-3.15
Sharpe ratio	2.68	2.5	2.33	1.87	1.06	-0.44
Hit rate (%)	21.9	21.51	21.37	20.69	20.69	41.87
Num. of trades	41729	41615	41189	39585	31657	400
Average Profit (USD)	1601	1581	1561	1505	1428	8731
Average Loss (USD)	283	282	286	288	313	7657
$\frac{\text{Avg. Profit}}{\text{Avg. Loss}}$	5.65	5.62	5.49	5.22	4.57	1.14

Table 7.4: Results (averages of 50 runs) of trading simulation over 2016. Sharpe Ratio is estimated as a monthly return. Both Avg. profit and Avg. loss are estimated per trade.



(a) Profits

(b) Sharpe

Figure 7.5: Distribution over 50 runs of (a) Profit and (b) Sharpe ratios in 2016

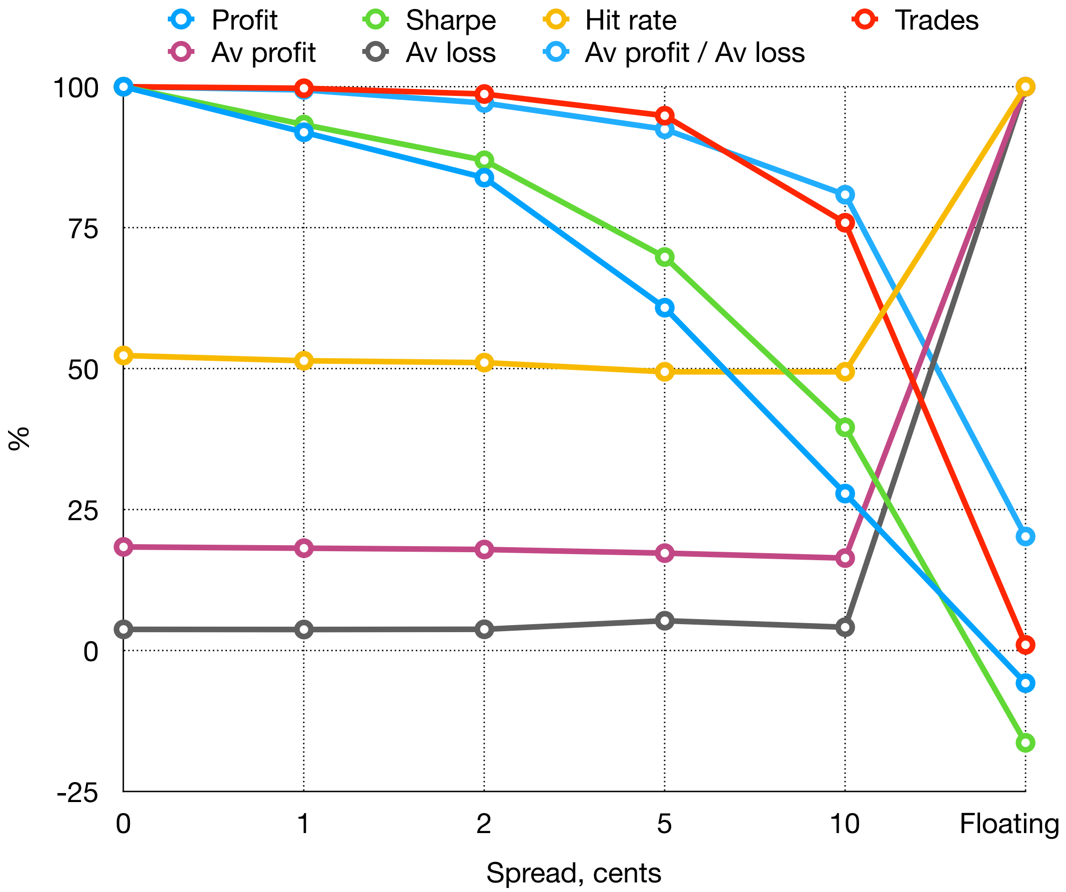
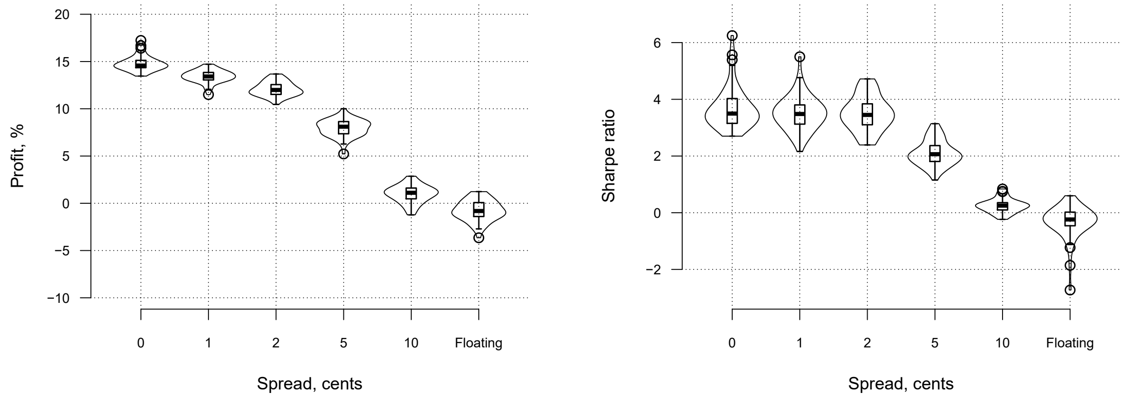


Figure 7.6: Normalized metrics 2016

Metric	Spread, USD					
	0	0.01	0.02	0.05	0.1	floating
Profit (%)	14.78	13.43	12.1	7.97	0.98	-0.765
Sharpe ratio	3.69	3.51	3.51	2.1	0.25	-0.31
Hit rate (%)	20.37	20.03	19.65	19.58	21.3	43.93
Num. of trades	22681	22370	22317	19723	10367	260
Average profit (USD)	1001	1007	1004	1039	1103	6361
Average loss (USD)	174	177	178	203	288	5501
$\frac{\text{Avg. profit}}{\text{Avg. Loss}}$	5.75	5.68	5.64	5.12	3.83	1.16

Table 7.5: Results (averages of 50 runs) of trading simulation over 2017. Sharpe Ratio is estimated as a monthly return. Both Avg. profit and Avg. loss are estimated per trade.



(a) Profits (b) Sharpe

Figure 7.7: Distribution over 50 runs of (a) Profit and (b) Sharpe ratios in 2017

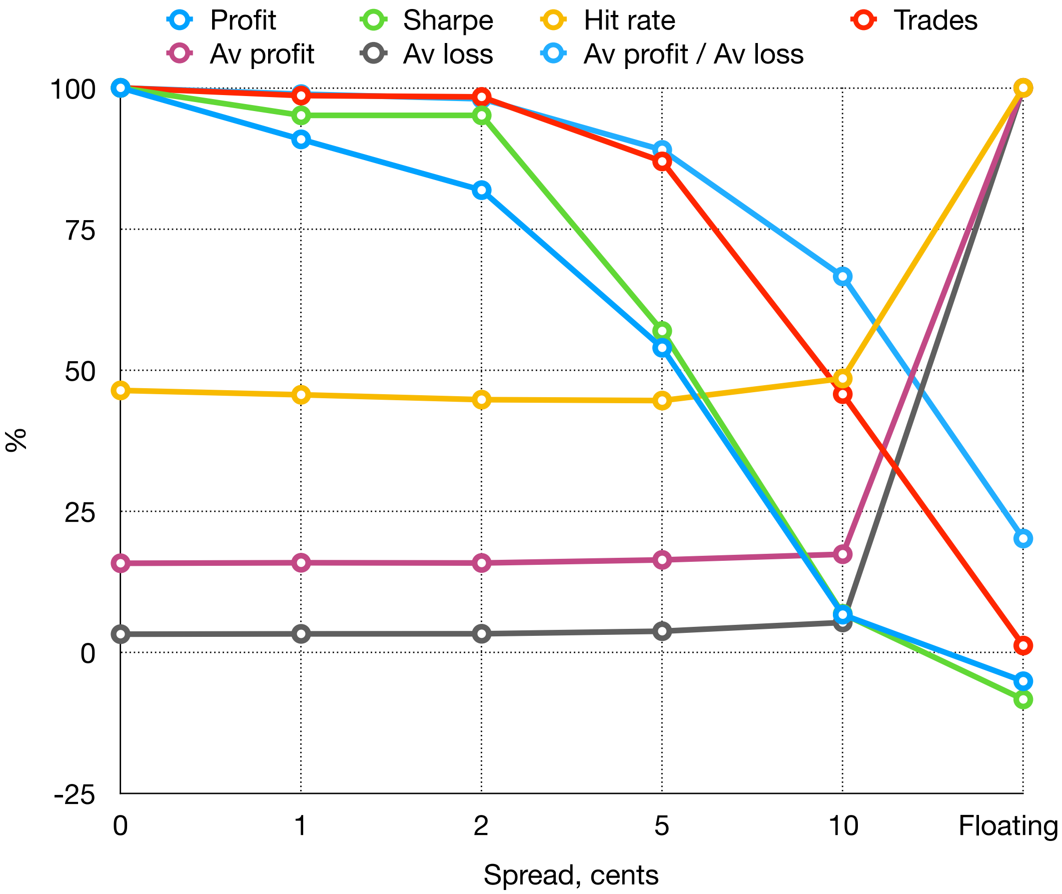


Figure 7.8: Normalized metrics 2017

7.5 Discussion

The research, as presented in this chapter, investigated the influence of hidden trading costs in the form of bid-ask spread on the effectiveness of a GP based intraday automated trading system. In total, 50 simulation runs were performed for each of the five fixed spreads and a floating spread over three years (2015, 2016 and 2017). The obtained results clearly show that nature and the size of the bid-ask spread dramatically affect the performance of an intraday automated trading system. As such these factors can be considered as one of the key influences that define the success or failure of a such system.

It has been shown that the floating spread with a median value of 0.02 USD results in a much worse performance than a fixed spread of 0.1 USD, i.e. an order of magnitude difference. These results appear to also support the observation that uninformed traders — as in traders that trade for liquidity needs — will be more susceptible to the negative impact of spreads [27]. Future work could repeat the analysis with data from a stock market with historically less volatility (e.g. NYSE rather than the NASDAQ). Moreover, we note that as fixed spreads are actually preferable to an automated trading agent, brokers have the opportunity to offer a fixed spread service while also maintaining a profitable service. However, an investigation of the parameterization of the fixed spread scenario to maintain a profitable service is left to future research.

Chapter 8

Conclusion and Future Work

Previous research established an approach for symbiotically coevolving both decision trees (DT) and technical indicators (TI) simultaneously [95, 97, 98]. Thus, each trading agent is the result of a unique interaction between TI (temporal features) and DT (decision tree) as coevolved in response to the specific current market situation. The symbiotic relation links the fitness expressed at the level of DT to the TI without having to define any surrogate performance functions for the TI. Specific recommendations from this earlier body of research included the use of use of criteria specifically designed to detect the onset of poor trading behavior [98] and the complete reinitialization of the DT–TI population before rebuilding trading agents [97]. The results of this earlier work can be summarized as follows:

- The use of criteria specifically designed to detect the onset of poor trading behavior, and therefore trigger the identification of a new trading agent, was demonstrated to be significantly more effective than the use of retraining intervals (with or without continuous evolution). This appears to be a particularly important factor for improving the quality of results (Section 3.1). In addition, the worst performance of the Static case confirms the non-stationary nature of the task.

- Reinitialization of the DT–TI population before retraining improves the performance compared to continuous evolution (Section 3.1). This illustrates the importance of maintaining the diversity of the DT–TI population. This result runs against current practice in which continuous evolution of a population initialized once is the norm.

- The use of the validation partition to test the DT–TI population quality and to select the best trading agent (if the population has passed the test) is the second important way to get better results, i.e., exploration appears to play a more significant role than exploitation.

This thesis began by revisiting the original Base FXGP framework, beginning with analysis of its weak points (Section 3.2) and its further development (Section 3.3). This provides the core of the FXGP framework assumed in this thesis.

Section 3.3 demonstrated that teams of champion trading agents improves the negative spread of runs compared to that of a single trading agent. At the same time the computational cost for maintaining a team of champion agents is still significantly $\approx 40\%$ lower than that for the original FXGP (e.g. [98]). The use of evolved teams demonstrated the best results in all categories: trustability (the percentage of profitable runs) and quartile scores. Indeed, this configuration provides statistically significant improvements over the single agent mode (95th percentile). In addition, it was during the evaluation of the proposed FXGP algorithm that the importance of the nature/type of spread (floating vs. fixed) became a significant factor for the performance of automated trading systems.

The results of experiments in Section 3.1 were shadowed by the use of the "abstract" average fixed spread for all three currency pairs. This issue was revisited in Section 3.3 with the historical prices containing the real Bid-Ask spread information.

Section 3.4 demonstrated the significance of coevolving DT and TI under foreign exchange data with candlesticks constructed over 30 minute intervals. Specifically, the FXGP algorithm was compared to the same framework, but limited to evolving DT alone. In the latter case, rather than evolve temporal features (TI) a set of popular TIs is assumed. The resulting comparison demonstrated that the FXGP algorithm significantly outperforms the benchmarking framework based on traditional TIs, i.e. it is limited to evolving decision trees alone. This is important because research using GP to evolve automated trading systems to date does not attempt to identify both TI and DT, thus further validating prior research [98].

The application of machine learning to non-stationary streaming data has motivated an independent body of research on prequential classification. With this in mind, benchmarking was performed to validate the FXGP algorithm against a task outside from financial data, that of predicting electricity utilization **(Chapter 4)**. This is potentially important because decision tree induction under streaming data has been a re-occurring theme for a considerable period of time. Specifically, the Hoeffding Tree method characterizes stream content statistically and relates this information to change detection. This forms the basis for decision tree methods that dynamically modify the structure of a classifier when exposed to stream content. Based on the results, the following conclusions can be made:

- GP can be applied to streaming data classification tasks and remain computationally feasible without recourse to specialist hardware/software support (e.g. no use was made of of custom hardware or GPUs), while the performance of the classifier is competitive with that of algorithms specifically designed for decision tree induction under streaming data.
- Preprocessing streaming data using a candlestick representation can be effective for non-financial data. Naturally, such a representation assumes that there is sufficient data present in the stream for construction of each candlestick. If the data was only available at, say, 30 minute intervals and this was also the rate at which predictions were required, then preprocessing using candlesticks would not be appropriate.

In review, the comparison between coevolved TI-DT with Hoeffding Tree approaches to streaming classification stripped away the components of FXGP that were specific to the automatic trading agent scenario (SL, TP, and retracement). This removed a significant number of parameters from the FXGP algorithm and clarified the contribution of coevolving TI with DT. In short, the explicitly evolutionary components of the FXGP algorithm were demonstrated to be competitive with currently well established methods.

Chapter 5 examined the use of three different techniques for identifying retracement opportunities in the FXGP framework, namely Moving average, Pivot points, and Fibonacci ratios. In summary, the limit (SL and TP) order verification and adjustment with support and resistance levels significantly improves overall profitability and helps address the challenges listed in Chapter 5.

The strategies based on the Fibonacci series clearly had the best performance of the three techniques. Moreover, Fibonacci retracement with 0% and 100% levels drawn through the Close prices (FXGPF) appeared to be more efficient than

Fibonacci retracement with 0% and 100% levels drawn through the High and Low prices, at least in case of FXGP. The use of Fibonacci retracement to define the SL order position increased the median score by 86.7%, increased the percent of profitable trades by 29.7%, and reduced the average size of a SL order by 51% compared to the FXGP without orders verification. The use of Fibonacci retracement to define the SL order position reduced the retraining time of the team of three trading agents by 12%. This reduction likely occurs because the use of Fibonacci levels to define the SL order position and increases the likelihood of a DT–TI population passing the validation process. Of the different modes of FXGP, FXGPF had the biggest average number of trades per simulated period of time and the lowest percent of triggered SL orders. Teams of trading agents were confirmed to be more efficient than single trading agents.

Due to a significant amount of time and resources required to perform simulations. The proposed approach was tested on only one currency pair and within a limited time frame. To make future conclusions about its applicability for different trading assets and markets, future work can include a wider set of trading assets and more recent price data. The development of a robust algorithm for Fibonacci retracement base level (0% and 100%) detection is also of interest. Moreover, the use of a genetic programming based approach is as much a personal choice, based on familiarity with the method. The observations from Chapters 3 and 5 are used to support the hypothesis that the FXGP formulation is capable of identifying profitable trading strategies. The remaining two chapters therefore attempt to build on this to highlight the impact of attempting to operate under more demanding market conditions, i.e. intraday trading with one minute price candlesticks.

The research of **Chapter 6** addresses challenges to technical analysis in general and automated trading in particular utilizing a multi-asset approach (challenges 1, 2 and 3) and frequent intraday trading with short (1 minute) time intervals (challenge 3). Results are presented that demonstrate a clear preference for adopting a ranking based on either a simple Moving Average or the Moving Sharpe Ratio (with Moving Sharpe Ratio showing better results over a simple Moving Average of Daily Returns in all cases), both in terms of the profitability and the Sharpe Ratio. Moreover, both schemes perform significantly better than the Kelly Criterion (a popular method for long-term investment [135, 106, 26, 87]). The Moving Sharpe Ratio outperforms other stock prioritizing heuristics for frequent intraday trading (with FXGP estimating the returns). In comparison, adopting a buy-and-hold investment strategy for all 86 stocks in the portfolio[1] resulted in significantly worse results. In short, even though the frequent intraday trading scenario incurs a transaction cost per trade, and there are typically tens of thousands of trades, only 2 of 20 parameterizations perform worse than the buy-and-hold strategy in 2015 and all perform better than buy-and-hold in 2016. Only 5 of 20 parameterizations perform worse than the buy-and-hold strategy in the 8 months of 2017. There are no parameterizations for which a stock ranking performed using the Kelly Criterion approaches that of either the Moving Average or the Moving Sharpe Ratio.

In addition, a general bias towards trading with single stocks per day was evident. That is to say, although increasing the number of stocks traded per day resulted in a lower 'average loss per trade' the corresponding 'average profit per trade' was also much lower. Thus, the resulting Sharpe Ratio (a measure of average profit-to-risk) was actually preferable for the case of performing frequent intraday trading with one stock per day. It is a mark of the accuracy of the estimated returns from the GP trading agents that such specific recommendations could be made without having a detrimental effect on the performance of the investment strategy, i.e. risk.

The experiments performed in Chapter 5 assumed a fixed spread of 1 cent. Such an assumption is not realistic, but was adopted in this case because the main goal was to evaluate the relative performance of the proposed asset selection algorithms and as compared to a well known traditional approach (buy-and-hold). The future work may include the real fixed (normally 2, 5, 10 cents and higher individual for each stock) and floating spreads and different markets.

Chapter 7 investigated the influence of hidden trading costs in the form of bid-ask spread on the effectiveness of a GP based intraday automated trading system, where this effect was first noticed during the evaluation of the FXGP algorithm in Section 3.3. FXGP is again deployed on the same subset of NASDAQ stock from the portfolio experiment where the NASDAQ is known to have more volatility than other stock markets such as the NYSE. It is empirically demonstrated that the nature and

---

[1] All stocks from the portfolio are held, as there is no basis for selecting a subset of specific stocks.

the size of the bid-ask spread dramatically affects the performance of the intraday automated trading system. As such these factors can be considered as one of the key influences defining the success or failure of a such system.

In addition, it was observed that the floating spread with a median value of 0.02 USD results in a much worse performance than a fixed spread of 0.1 USD, i.e., an order of magnitude difference. This can be explained by recognizing that a floating spread represents a continuously changing trading environment. As a consequence, the trading conditions under which the GP algorithm is trained and then forced to trade are different, and it is this that results in a significant degradation of its performance. Therefore, it is crucial to choose a broker with fixed spreads for automated trading algorithms using genetic programming and possibly other machine learning techniques. These results appear to also support the observation that uninformed traders – as in traders that trade for liquidity needs – will be more susceptible to the negative impact of spreads ([27]).

In summary, the trading conditions are integral to and one of the most important parts of any trading system. Understanding their role, especially the role of the hidden trading costs such as Bid-Ask spread, is essential for building a profitable trading algorithm. As such, addressing the influence of hidden trading costs can help to reduce the impact of all the challenges outlined in Chapter 1. This thesis systematically investigated the impact of floating and fixed spreads, demonstrating that fixed spreads that are an order of magnitude larger are potentially preferable to a floating spread. How a broker might use this to, on the one hand, attract automated trading agents while also maximizing service revenue is left for future research.

Future work could repeat the analysis with data from different stock markets (e.g. NYSE rather than the NASDAQ) and data from brokers that offer fixed stock spreads. Algorithm modifications supporting profitability of intraday trading on 1-minute intervals would be of particular interest. In comparison, current state of the art in financial forecasting assume 30-minute intervals [12], which is sufficient for informing interday trading, but not automated trading under intraday trading conditions.

Appendix A

Skewness and Kurtosis of NASDAQ 100 stocks

Stock	Skewness 2015	Kurtosis 2015	Skewness 2016	Kurtosis 2016	Skewness 2017	Kurtosis 2017
AAL	0.4937	2.2027	0.1395	2.8054	0.1373	2.4184
AAPL	-0.2039	1.7290	0.0457	1.7177	-0.6536	2.8716
ADBE	0.5141	2.4686	-0.5685	3.0411	-0.3523	1.9521
ADI	0.2769	2.3129	0.4916	2.8514	-0.5054	2.9175
ADP	-0.7786	3.0167	0.4986	3.8358	1.4456	6.9099
ADSK	-0.4453	2.3183	0.1776	2.2629	0.1394	1.3875
AKAM	-0.8726	2.5455	0.6736	3.0457	0.2039	1.4965
ALXN	0.0281	2.9833	0.9156	4.0237	-0.8057	3.4714
AMAT	0.2269	1.8512	-0.0732	1.6965	-0.2861	1.8618
AMGN	-0.3517	4.3520	0.3439	1.9848	-0.0299	2.0572
AMZN	0.3713	2.1593	-0.5269	2.0701	-0.2191	1.7754
ATVI	0.6833	2.3256	-0.1609	2.1701	-0.3732	1.9276
BIDU	-0.8594	2.5110	0.0782	2.7708	1.4803	3.8632
BIIB	-0.0320	1.4923	-0.0311	2.0426	-0.0094	2.2902
BMRN	16.3986	269.9169	-0.2147	3.1455	-0.0325	3.4423
CA	-0.0744	1.8038	-0.6908	2.9426	1.1711	3.8793
CELG	0.8215	3.6072	0.7062	2.6236	0.2933	1.9964
CERN	-0.1105	1.9061	0.1306	2.1967	-0.4826	1.9341
CHKP	0.0710	2.2264	-0.1379	1.8535	-0.8273	3.8040
CMCSA	0.1888	2.4939	-0.3510	2.3637	0.0174	2.4698
CSCO	-0.4609	2.6283	-0.9057	3.1202	0.2437	1.7901
CSX	-0.2488	1.5788	0.8420	3.0423	-1.2743	5.1646
CTAS	0.1102	2.3000	0.1629	1.5621	0.2970	2.2291

Continued on next page

Table A.1 – continued from previous page

Stock	Skewness 2015	Kurtosis 2015	Skewness 2016	Kurtosis 2016	Skewness 2017	Kurtosis 2017
CTRP	0.4869	2.6558	-0.1436	2.4859	-0.1725	1.9664
CTSH	-0.6731	3.8113	-0.4972	2.6783	0.0322	1.6294
CTXS	0.2604	2.1329	-1.0652	3.2578	-0.4658	2.5573
DISCK	-0.4365	2.0510	-0.0466	2.0687	-0.8977	3.1996
DISH	-0.0209	1.9297	-0.0707	2.2317	-0.6453	2.8687
DLTR	-0.6667	2.1398	0.6918	2.1098	-0.5640	2.6637
EA	-0.5353	2.4209	-0.4268	1.8390	0.0339	1.4065
EBAY	0.2452	1.9148	0.2970	1.5250	-0.6372	3.2989
ESRX	0.2800	2.6491	0.9205	4.3922	0.5152	2.0691
FAST	0.0871	3.1701	-0.3539	2.2576	0.2868	1.5113
FB	0.4918	1.9721	-0.5081	2.8516	0.1612	2.2414
FOX	-0.6119	2.1361	-0.0386	2.1812	0.1718	1.7072
FOXA	-0.4874	1.9006	-0.0816	2.2299	0.1290	1.6948
GILD	0.6087	2.2987	0.4097	2.3984	0.5164	3.0622
HAS	-0.7341	2.4960	-0.8148	3.1568	-0.4415	2.9069
HOLX	-0.5112	2.3469	0.0567	1.7356	-0.2456	1.8703
HSIC	0.7283	2.9393	-0.1067	1.9890	-0.2881	2.2160
IDXX	-0.1777	1.9723	-0.0531	1.5758	-1.1527	3.4775
ILMN	-0.6382	3.2497	0.3124	1.8859	-0.2349	4.1568
INCY	-0.6505	2.7652	0.4852	2.4277	0.1283	3.1007
INTC	-0.1898	2.2069	-0.0092	1.8523	-0.3609	2.8027
INTU	-0.3579	2.0112	-0.4237	2.3792	0.1574	1.5247
ISRG	-0.0744	3.3155	-0.4967	2.3454	-0.1395	1.5899
JBHT	-0.1138	1.9341	0.3951	3.9539	-0.0301	1.9009
KLAC	-0.1266	1.8588	0.2764	2.6239	-0.3083	2.6769
LBTYK	-0.4520	2.3561	-0.6415	3.4798	-0.4820	2.1485
LRCX	-0.7840	3.1710	0.1019	2.0264	-0.1994	1.5252

Continued on next page

Table A.1 – continued from previous page

Stock	Skewness 2015	Kurtosis 2015	Skewness 2016	Kurtosis 2016	Skewness 2017	Kurtosis 2017
MAR	-0.2597	2.2576	0.9388	4.1423	-0.1906	1.6100
MAT	0.1081	2.4980	-1.1741	4.1692	0.0838	2.7244
MCHP	-0.3588	2.2879	0.0048	1.6892	-0.3407	2.2469
MDLZ	-0.1111	1.7282	-0.7326	3.7706	-0.1069	4.0174
MSFT	0.8367	2.6990	0.2963	2.0573	0.1515	1.6540
MU	0.1473	1.4168	0.5658	2.2375	-0.3304	1.9445
MXIM	0.8447	3.0327	-0.5538	2.4231	-1.0953	5.4731
NCLH	-0.6722	2.5631	0.4763	2.2921	0.0047	2.2773
NFLX	-0.2795	1.9282	0.8533	2.5783	0.6714	2.7557
NTES	0.5177	2.5209	0.2633	1.7479	-0.4920	2.8003
NVDA	1.2262	3.2045	0.8016	2.9065	0.3103	1.3960
ORLY	-0.1897	2.2794	-0.8726	3.5769	-0.6440	2.0463
PAYX	0.3898	2.6704	-0.2921	2.2126	0.0441	2.2046
PCAR	-0.8267	2.2865	-0.0114	3.2392	-0.3640	2.0360
PCLN	-0.0972	3.2997	-0.6303	3.0208	-0.3519	2.5801
QCOM	-0.4877	1.9772	0.0431	1.6705	1.4537	5.2046
ROST	0.0164	2.1958	0.3016	1.9799	-0.5425	1.9119
SBUX	-0.2835	2.0964	0.1421	2.2952	0.2706	2.6016
SHPG	-0.1907	1.7939	-0.1632	2.0893	-0.8391	3.4261
SIRI	-0.6186	3.6434	-0.0115	2.9828	-0.1027	2.2129
STX	-0.3876	2.2016	-0.5663	2.3169	-0.5176	2.2078
SWKS	0.2584	2.0219	-0.2719	1.8760	-1.5139	5.2746
SYMC	-0.1391	1.5457	0.0656	1.5235	-0.8064	3.8018
TSCO	-0.6699	3.5137	-0.5996	2.0924	0.1200	1.4521
TSLA	0.0680	1.9133	-0.1378	3.4323	-0.0979	1.8211
TXN	-0.5376	2.1951	-0.2097	1.7215	-0.7202	2.5903
ULTA	-0.2023	2.7413	-0.5715	2.0507	-0.7881	3.8707

Continued on next page

Table A.1 – continued from previous page

Stock	Skewness 2015	Kurtosis 2015	Skewness 2016	Kurtosis 2016	Skewness 2017	Kurtosis 2017
VIAB	-0.3189	1.4452	0.0587	2.1217	-0.1474	1.8090
VOD	0.2768	2.1390	-0.7746	2.8448	0.0288	1.4825
VRSK	-0.5912	3.3785	-1.0157	3.5749	0.4393	2.9026
VRTX	-0.3797	3.0387	1.0318	5.4493	0.2287	1.9336
WBA	-0.1023	3.0741	-0.2078	3.3044	-0.0838	2.3811
WDC	-0.2226	1.9513	0.6685	2.8717	-0.2199	1.9889
WYNN	0.1770	1.7693	-1.5115	4.6062	-0.4109	1.7099
XLNX	0.2677	1.9286	0.6144	3.0976	0.0757	1.6460
XRAY	1.0834	2.7328	-0.2197	2.8914	-1.0025	2.7336

Table A.1: The annual Skewness and Kurtosis metrics of NASDAQ 100 stocks

Appendix B

Sample Decision Trees

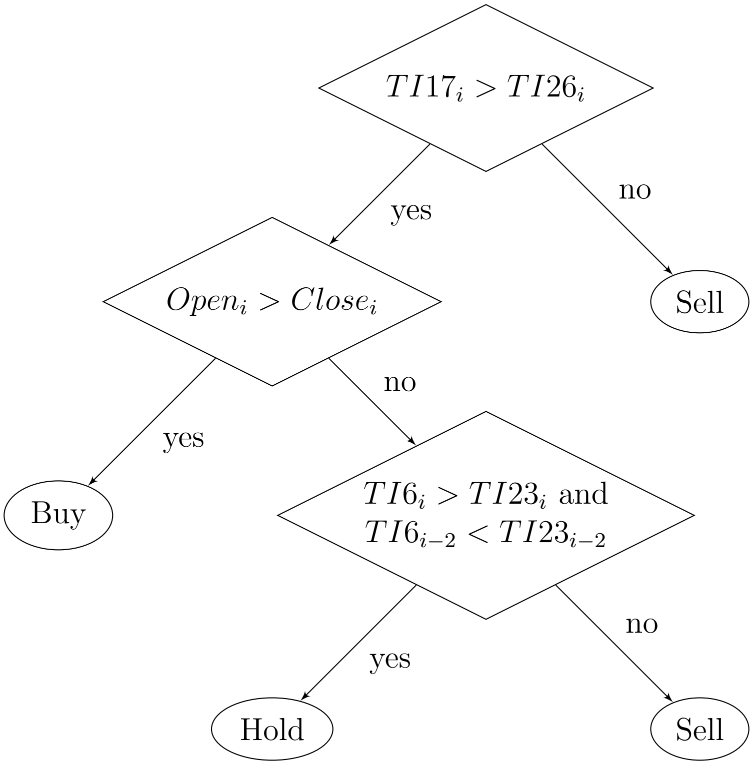


Figure B.1: Sample DT (Base FXGP, Section 3.1), Trades: 5

where:

- $Open_i$, $High_i$, Low_i and $Close_i$ are, respectively, Open, High, Low, and Close prices at moment i.
- $TI6 = MA_{24}(Low_{i-5})$
- $TI17 = MA_{64}(Close_{i-1})$
- $TI23 = MA_{44}(Low_i)$
- $TI26 = MA_{24}(High_i)$

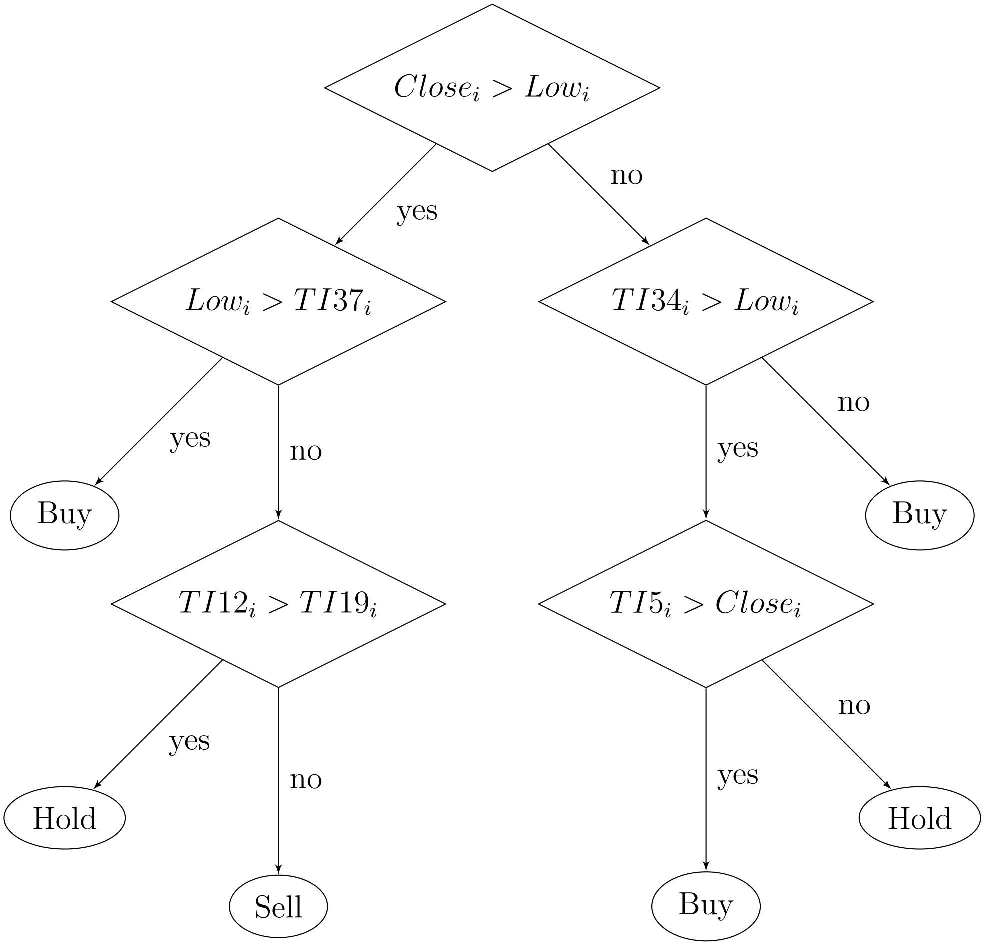


Figure B.2: Sample DT (FXGP, Section 3.3), Trades: 3

where:

- $Open_i$, $High_i$, Low_i and $Close_i$ are, respectively, Open, High, Low, and Close prices at moment i.
- $TI5 = MA_2(High_i)$
- $TI12 = MA_7(Low_i)$
- $TI19 = MA_{30}(Open_i)$
- $TI34 = MA_{22}(Close_i - Close_{i-1} + High_i)$
- $TI37 = MA_7(Open_i)$

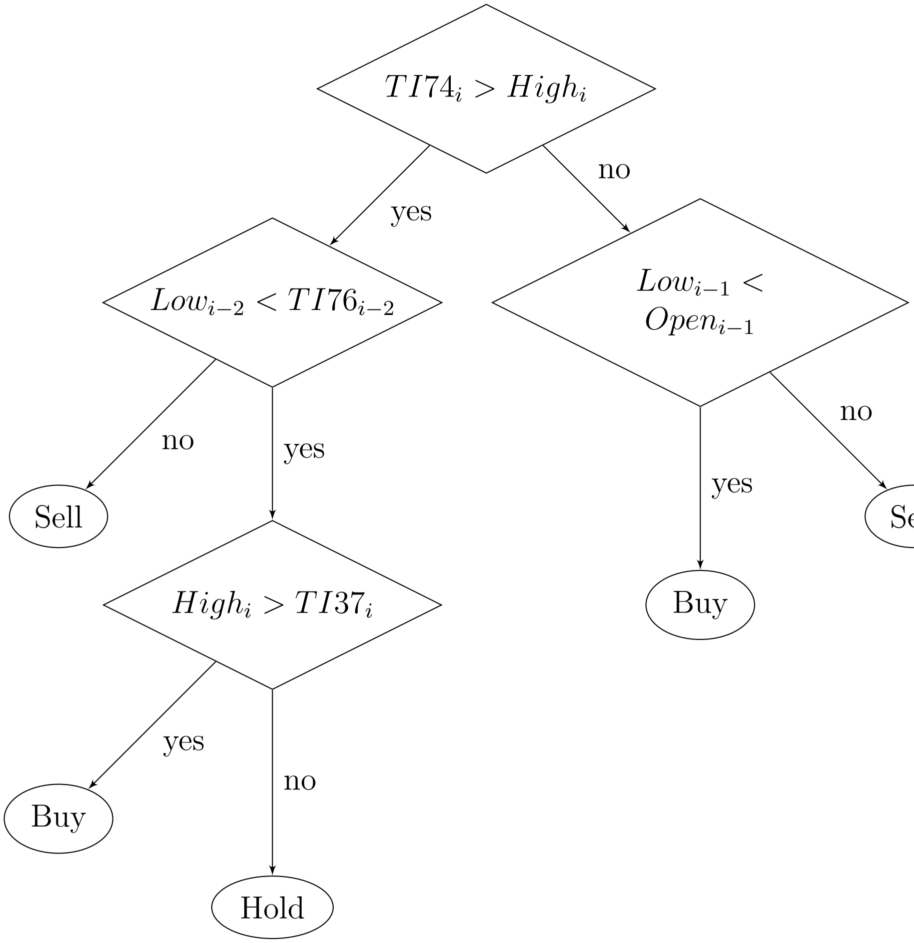


Figure B.3: Sample DT (FXGP, Section 3.3), Trades: 18

where:

- $Open_i$, $High_i$, Low_i and $Close_i$ are, respectively, Open, High, Low, and Close prices at moment i.

- $TI37 = MA_{54}(High_i - Open_i + Close_1)$

- $TI74 = MA_{90}(Low_{i-3} + (Open_i - Close_1)/2)$

- $TI76 = MA_{86}(Low_{i-3} + (Open_i - Close_1)/2)$

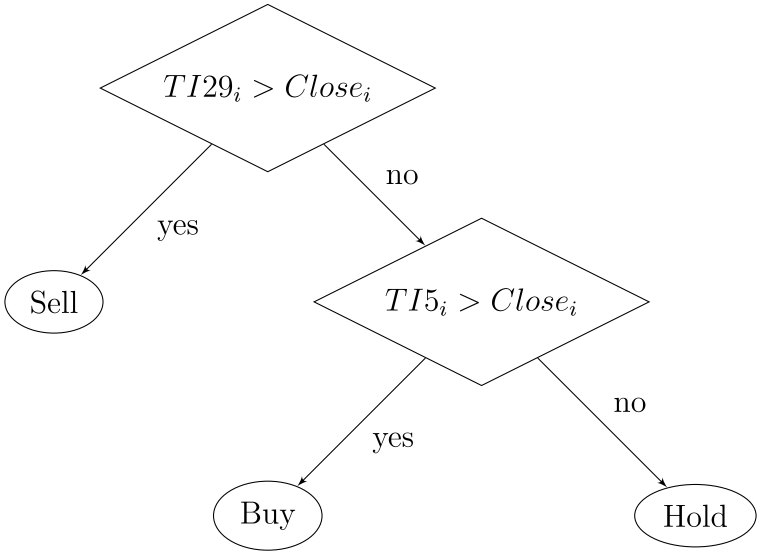


Figure B.4: Sample DT (Base FXGP, Section 3.1), Trades: 32

where:

- $Open_i$, $High_i$, Low_i and $Close_i$ are, respectively, Open, High, Low, and Close prices at moment i.
- $TI5 = MA_{14}(Close_i + (High_i - Close_{i-2})/2)$
- $TI29 = MA_{18}(Low_i)$

All moving averages are calculated as (3.1).

Bibliography

[1] Nidhi Aggarwal and Susan Thomas. The causal impact of algorithmic trading on market quality. *Indira Gandhi Institute of Development Research*, July:36, 2014.

[2] Irene Aldridge. *High-frequency trading: a practical guide to algorithmic strategies and trading systems*. Wiley Trading, 2010.

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

[4] Yakov Amihud and Haim Mendelson. Asset pricing and the bid-ask spread. *Journal of Financial Economics*, 17:223–249, 1986.

[5] James J. Angel and Douglas McCabe. Fairness in financial markets: The case of high frequency trading. *Journal of Business Ethics*, 112:585–595, 2013.

[6] P. J. Angeline. Subtree crossover causes bloat. *Genetic Programming 1998: Proc. 3rd Annu. Conf. (GP98)*, pages 745–752, 1998.

[7] A. Atwater, M. I. Heywood, and A. N. Zincir-Heywood. GP under streaming data constraints: A case for Pareto archiving? In *ACM GECCO*, pages 703–710, 2012.

[8] R. Azencott, A. Beri, Y. Gadhyan, and N. Joseph. Real-time market microstructure analysis: online transaction cost analysis. *Quantitative Finance*, 14:1167–1185, 2014.

[9] F. M. Bandi and J. R. Russell. Microstructure noise, realized variance, and optimal sampling. *Review of Financial Studies*, 75:339–369, 2008.

[10] W. Banzhaf, F. D. Francone, and P. Nordin. The effect of extencive use of the mutation operator on the generalization in genetic programming using sparse data sets. *4th International Conference on Parallel Problim Solving from Nature (PPSN96)*, pages 300–309, 1996.

[11] Matthew Baron, Jonathan Brogaard, Bjorn Hagstromer, and Andrei Kirilenko. Risk and return in high-frequency trading. *Journal of Financial and Quantitative Analysis*, November:82, 2014.

[12] S. Barra, S. Carta, A., A. Podda, and D. Recupero. Deep learning and time series-to-image encoding for financial forecasting. *IEEE/CAA Journal of Automatica Sinica*, 7:683–692, 2020.

[13] J. Baz and H. Guo. An asset allocation primer: Connecting Markowitz, Kelly and Risk Parity. *Quantative Research*, pages 1–16, 2017.

[14] Hendrik Bessembinder and Kumar Venkataraman. Bid-ask spreads: Measuring trade execution costs in financial markets. *Encyclopedia of Quantitative Finance*, May, 2010.

[15] A. Bifet. *Adaptive Stream Mining: Pattern Learning and Mining from Evolving Data Streams*, volume 207 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2010.

[16] A. Bifet, I. Žliobaitė, B. Pfahringer, and G. Holmes. Pitfalls in benchmarking data stream classification and how to avoid them. In *Machine Learning and Knowledge Discovery in Databases*, volume 8188 of *LNCS*, pages 465–479, 2013.

[17] Albert Bifet. MOA: Massive Online Analysis Learning Examples. *Journal of Machine Learning Research*, 11:1601–1604, 2010.

[18] Zsolt Bitvai and Trevor Cohn. Day trading profit maximization with multi-task learning and technical analysis. *Machine Learning*, page 23, 2014.

[19] Graham Bowley. Preserving a market symbol. *The New York Times*, 2011.

[20] Anthony Brabazon, Michael Kampouridis, and Michael O'Neill. Applications of genetic programming to finance and economics: past, present, future. *Genetic Programming and Evolvable Machines*, 21(1):33–53, 2020.

[21] M. Brameier and W. Banzhaf. *Linear Genetic Programming*. Springer, 2007.

[22] Henry L. Bryant and Michael S. Haigh. Bid-ask spreads in commodity futures markets. *Applied Financial Economics*, 14(13):923–936, 2004.

[23] Matthew Butler and Dimitar Kazakov. Particle swarm optimization of bollinger bands. In *International Conference on Swarm Intelligence*, volume 6234 of *LNCS*, pages 504–511, 2010.

[24] Matthew Butler and Dimitar Kazakov. A learning adaptive Bollinger band system. In *IEEE Conference on Computational Intelligence on Financial Engineering & Economics*, pages 1–8, 2012.

[25] M. Castelli, L. Vanneschi, and M. De Felice. Forecasting short-term electricity consumption using a semantics-based genetic programming framework: The south italy case. *Energy Economics*, 47:37–41, 2015.

[26] Ernest P. Chan. *Quantitative trading: how to build your own algorithmic trading business*. John Wiley and Sons, 2009.

[27] Yuk-Shee Chan and Mark Weinstein. Reputation, Bid-Ask Spread and Market Structure. *Financial Analysts Journal*, 49(4):57–62, 2006.

[28] Kumar Chellapilla. Evolving computer programs without subtree crossover. *IEEE Transactions on Evolutionary Computation*, 1:209–216, 1997.

[29] S. Chen and C. Yeh. Toward a computable approach to the efficient market hypothesis: an application of genetic programming. *Journal of Economic Dynamics and Control*, 21, 1996.

[30] Simone Cirillo, Stefan Lloyd, and Peter Nordin. Evolving intraday foreign exchange trading strategies utilizing multiple instruments price series. *arXiv e-prints*, 1411.2153, 2014.

[31] Robert W Colby. *The encyclopedia of technical market indicators*. McGraw-Hill, 2003.

[32] I. Contreras, J. I. Hidalgo, L. Nuñez-Letamendía, and J. M. Velasco. A meta-grammatical evolutionary process for portfolio selection and trading. *Genetic Programming and Evolvable Machines*, 18(4):411–431, 2017.

[33] G. Creamer. Model calibration and automated trading agent for Euro futures. *Quantitative Finance*, 12(4):531–545, 2012.

[34] G. Creamer and Y. Freund. Automated trading with boosting and expert weighting. *Quantitative Finance*, 10(4):401–420, 2010.

[35] W. Cui, A. Brabazon, and M. O'Neill. Adaptive trade execution using a grammatical evolution approach. *International Journal of Financial Markets and Derivatives*, 2(1/2):4–31, 2011.

[36] M. Cunkas and M. Taskiran. Turkey's electricity consumption forecasting using genetic programming. *Energy Sources, Part B: Economics, Planning and Policy*, 6:406–416, 2011.

[37] Puja Das, Nicholas Johnson, and Arindam Banerjee. Online lazy updates for portfolio selection with transaction costs. *Proceedings of the 27th AAAI Conference on Artificial Intelligence, AAAI 2013*, pages 202–208, 2013.

[38] V. DeMiguel, L. Garlappi, and R. Uppal. Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy? *Review of Financial Studies*, 22:1915–1953, 2009.

[39] I. Dempsey, M. O'Neill, and A. Brabazon. Adaptive trading with grammatical evolution. In *IEEE Congress on Evolutionary Computation*, pages 2587–2592, 2006.

[40] I. Dempsey, M. O'Neill, and A. Brabazon. *Foundations in Grammatical Evolution for Dynamic Environments*, volume 194 of *Studies in Computational Intelligence*. Springer, 2009.

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

[42] Harold Demsetz. The cost of transacting. *The Quarterly Journal of Economics*, 82, 1968.

[43] G. Ditzler and R. Polikar. Incremental learning of concept drift from streaming balanced data. *IEEE Transactions on Knowledge and Data Engineering*, 25(10):2283–2301, 2013.

[44] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar. Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, 10(4):12–25, 2015.

[45] P. Domingos and G. Hulten. Mining high-speed data streams. In *ACM International Conference on Knowledge Discovery and Data Mining*, pages 71–80, 2000.

[46] J. A. Doucette, A. R. McIntyre, P. Lichodzijewski, and M. I. Heywood. Symbiotic coevolutionary genetic programming. *Genetic Programming and Evolvable Machines*, 13(1):71–101, 2012.

[47] C. L. Dunis, J. Laws, and G. Sermpinis. Higher order and recurrent neural architectures for trading the EUR/USD exchange rate. *Quantitative Finance*, 11(4):615–629, 2011.

[48] Ahmed Duran and Michael Bommarito. A profitable trading and risk management strategy despite transaction costs. *Quantitative Finance*, 11:829–848, 2011.

[49] Roger Edelen, Richard Evans, and Gregory Kadlec. Shedding light on ”invisible” costs: Trading costs and mutual fund performance. *Financial Analysts Journal*, 61, Issue 1:33–44, 2013.

[50] Robert D. Edwards, John Magee, and W.H.C. Bassetti. *Technical Analysis of Stock Trends*. Auerbach Publications, 2007.

[51] Edwin J. Elton, Martin J. Gruber, Stephen J. Brown, and William N. Goetzmann. *Modern portfolio theory and investment analysis*. Wiley, 2014.

[52] P. Fernandez-Blanco, D. Bodas-Sagi, F. Soltero, and J. Hidalgo. Technical market indicators optimization using evolutionary algorithms. In *ACM Conference Companion on Genetic and Evolutionary Computation*, pages 1851–1858, 2008.

[53] Asian Banking & Finance. `https://asianbankingandfinance.net/`.

[54] J. Gama. *Knowledge Discovery from Data Streams*. CRC Press, 2010.

[55] J. Gama, P. Medas, G. Castillo, and P. Rodrigues. Learning with drift detection. In *Proceedings of the Brazilian Symposium on Advances in Artificial Intelligence*, pages 286–295, 2004.

[56] Pedro Godinho. Can abnormal returns be earned on bandwidth-bounded currencies? evidence from a genetic algorithm. *Economic Issues*, 17:1–26, 2012.

[57] Peter Gomber, Kai Zimmerman, and Nenad Tomic. Algorithmic trading in practice. *The Oxford Handbook of Computational Economics and Finance*, February:311–332, 2018.

[58] Nikitas Goumatianos, Ioannis Christou, and Peter Lindgren. Stock selection system: Building long/short portfolios using intraday patterns. *Procedia Economics and Finance*, 5:298–307, 2013.

[59] G. N. Gregoriou. *Handbook of High Frequency Trading.* Academic Press, 2015.

[60] Justine Gregory-Williams, Bill Williams, and Bill Williams. *Trading chaos: maximize profits with proven technical techniques.* A marketplace book. J. Wiley, 2nd ed edition, 2004.

[61] Youngmin Ha. Algorithmic trading in limit order books for online portfolio selection. *SSRN Electronic Journal*, 2018.

[62] M. Harries. Splice-2 comparative evaluation: Electricity pricing. Technical report, University of New South Wales, 1999.

[63] M. I. Heywood. Evolutionary model building under streaming data for classification tasks: opportunities and challenges. *Genetic Programming and Evolvable Machines*, 2015. DOI `10.1007/s10710-014-9236-y`.

[64] A. Hirabayashi, C. Aranha, and H. Iba. Optimization of the trading rule in foreign exchange using genetic algorithm. In *ACM Conference on Genetic and Evolutionary Computation*, pages 1529–1536, 2009.

[65] Ulf Holmberg, Carl Lonnbark, and Christian Lundstrom. Assessing the profitability of intraday opening range breakout strategies. *Finance Research Letters*, September:11, 2012.

[66] A. Hryshko and T. Downs. An implementation of genetic algorithms as a basis for a trading system on the foreign exchange market. In *IEEE Congress on Evolutionary Computation*, pages 1695–1701, 2003.

[67] Bming Huang, Yuxiang Huan, Li Da Xu, Lirong Zheng, and Zhuo Zou. Automated trading systems statistical and machine learning methods and hardware implementation: a survey. *Enterprise Information Systems*, 13:132–144, 2019.

[68] Hiroshi Iba and Claus A. Aranha. *Practical Applications of Evolutionary Computation to Financial Engineering.* Springer, 2012.

[69] Investopedia, accessed Sept, 2012. `http://www.investopedia.com/terms/t/two-percent-rule.asp#axzz2710QU8jR`.

[70] Investopedia, accessed Sept, 2012. `http://www.investopedia.com/terms/n/noise.asp#axzz27d0d2rid`.

[71] R. Jagannathan and T. Ma. Risk reduction in large portfolios: Why imposing the wrong constraints helps. *Journal of Finance*, 58:1651–1684, 2003.

[72] N. Japkowicz and M. Shah. *Evaluating Learning Algorithms: A classification perspective*. Cambridge University Press, 2011.

[73] Christine X. Jiang, Jang Chul Kim, and Robert A. Wood. A comparison of volatility and bid-ask spread for NASDAQ and NYSE after decimalization. *Applied Economics*, 43(10):1227–1239, 2011.

[74] M. Kampouridis. An initial investigation of choice function hyper-heuristics for the problem of financial forecasting. In *IEEE Congress on Evolutionary Computation*, pages 2406–2413, 2013.

[75] M. Kampouridis and E. Tsang. EDDIE for investment opportunities forecasting: Extending the search space of the GP. In *IEEE Congress on Evolutionary Computation*, pages 2019–2026, 2010.

[76] Stephen Kelly, Jacob Newsted, Wolfgang Banzhaf., and Cedric Gondro. A modular memory framework for time series prediction. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2020.

[77] Sara Khanchi, Ali Vahdat, Malcolm Heywood, and Nur Zincir-Heywood. On botnet detection with genetic programming under streaming data label budgets and class imbalance. *Swarm and Evolutionary Computation*, 39, September 2017.

[78] G. Kim and S. Jung. The construction of the optimal investment portfolio using the Kelly Criterion. *World Journal of Social Sciences*, 3.6:15–26, 2013.

[79] Andrei A. Kirilenko, Albert S. Kyle, Mehrdad Samadi, and Tugkan Tuzun. The flash crash: The impact of high frequency trading on an electronic market. *SSRN Electronic Journal*, pages 1–42, 2012.

[80] Charles D. Kirkpatrick and Julie R. Dahlquist. *Technical analysis: the complete resource for financial market technicians*. FT Press, 2nd ed edition, 2011.

[81] Robert Kissell. *The science of algorithmic trading and portfolio management*. The science of algorithmic trading and portfolio management, 2014.

[82] Marek Andrzej Kocinski. On transaction costs in stock trading. *Quantative Methods In Economics*, XVIII, Issue 1:58–67, 2017.

[83] P. N. Kolm, R. Tütüncü, and F. J. Fabozzi. 60 years of portfolio optimization: Practical challenges and current trends. *European Journal of Operational Research*, 234:356–371, 2014.

[84] John R Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection v. 1 (Complex Adaptive Systems)*. MIT Press, 1993.

[85] M. Kritzman, S. Page, and D. Turkington. In defense of optimization: the fallacy of 1/n. *Financial Analysis*, 66:1–9, 2010.

[86] Justin Kuepper. *Money Management Using the Kelly Criterion*. Investopedia, 2004.

[87] Paolo Laureti, M. Medo, and Yi Cheng Zhang. Analysis of Kelly-optimal portfolios. *Quantitative Finance*, 10:689–697, 2010.

[88] J. Leung and T. Chong. An empirical comparison of moving average envelopes and bollinger bands. *Applied Economics Letters*, 10(6):339–341, 2003.

[89] Bin Li, Jialei Wang, Dingjiang Huang, and Steven Hoi. Transaction cost optimization for online portfolio selection. *Quantitative Finance*, 18:1411–1424, 2018.

[90] M. Lichman. UCI machine learning repository, 2013.

[91] P. Lichodzijewski and M. I. Heywood. Symbiosis, complexification and simplicity under GP. *ACM Genetic and Evolutionary Computation Conference*, 2010.

[92] K. Lien. *Day trading and swing trading the currency market: technical and fundamental strategies to profit from market moves*. John Wiley and Sons, Inc., 2008.

[93] Nick K. Lioudis. *Sharpe ratio*. Investopedia, 2017.

[94] Qianqiu Liu. On portfolio optimization: How and when do we benefit from high-frequency data? *Journal of Applied Econometrics*, 24:560–582, 2009.

[95] Alexander Loginov. On the utility of evolving forex market trading agents with criteria based retraining. Master's thesis, Dalhousie University, Faculty of Computer Science, March 2013.

[96] Alexander Loginov and Malcolm Heywood. On the different impacts of fixed versus floating bid-ask spreads on an automated intraday stock trading. *The North American Journal of Economics and Finance*, June 2020.

[97] Alexander Loginov and Malcolm I. Heywood. On the impact of streaming interface heuristics on GP trading agents: an FX benchmarking study. In *Proceeding of the ACM Genetic and Evolutionary Computation Conference*, pages 1341–1348, 2013.

[98] Alexander Loginov and Malcolm I. Heywood. On the Utility of Trading Criteria Based Retraining in Forex Markets. In *Applications of Evolutionary Computation (EvoFIN)*, volume 7835 of *LNCS*, pages 192–202, 2013.

[99] Alexander Loginov and Malcolm I. Heywood. On evolving multi-agent fx traders. In *Applications of Evolutionary Computation (EvoFIN)*, volume 8602 of *LNCS*, pages 203–214, 2014.

[100] Alexander Loginov, Garnett Wilson, and Malcolm I. Heywood. Better trade exits for foreign exchange currency trading using FXGP. In *IEEE Congress on Evolutionary Computation, CEC 2015, Sendai, Japan, May 25-28, 2015*, pages 2510–2517, 2015.

[101] Alexander Loginov, Garnett Wilson, and Malcolm I. Heywood. Benchmarking a coevolutionary streaming classifier under the individual household electric power consumption dataset. In *IEEE-INNS Joint Conference on Neural Networks – Special session track on Concept Drift, Domain Adaptation and Learning in Dynamic Environments*, 2016.

[102] Alexander Loginov, Garnett Wilson, and Malcolm I. Heywood. Stock selection heuristics for performing frequent intraday trading with genetic programming. *Genetic Programming and Evolvable Machines*, 2020.

[103] Otto Loistl, Bernd Schlossmann, Olaf Vetter, and Alexander Veverka. A comparison of transaction costs on Xetra and on NASDAQ. *Quantitative Finance*, 2:199–216, 2002.

[104] Tony Loton. *Stop orders: a practical guide to using stop orders for traders and investors*. Harriman House, 2009.

[105] Shingo Mabu, Kotaro Hirasawa, Masano Obayashi, and Takashi Kuremoto. Enhanced decision making mechanism of rule-based genetic network programming for creating stock trading signals. *Expert Systems with Applications*, 40:6311–6320, 2013.

[106] Leonard C. MacLean, Edward O. Thorp, Yonggan Zhao, and William T. Ziemba. How does the fortune's formula-kelly capital growth model perform? *The Journal of Portfolio Management*, 37:96–111, 2011.

[107] Rathinasamy Maheswaran and Rakesh Khosa. Multi resolution genetic programming approach for stream float forecasting. In *SEMCCO*, volume 7076 of *LNCS*, pages 714–722. Springer, 2011.

[108] J. Malek and T. V. Quang. Investing in high frequency data. *Business Trends*, 6.3:21–27, 2016.

[109] V. Manahov. The rise of the machines in commodities markets: new evidence obtained using strongly typed genetic programming. *Annuals of Operations Research*, 260:321–352, 2018.

[110] Viktor Manahov and Robert Hudson. New evidence of technical trading profitability. *Economics Bulletin*, 33(4):2493–2503, 2013.

[111] MarketsandMarkets^{TM}. Algorithmic trading market by trading type (forex, stock markets, etf, bonds, cryptocurrencies), component (solutions and services), deployment mode (cloud and on-premises), enterprise size, and region - global forecast to 2024. *MarketsandMarkets*^{TM}, 2019.

[112] H. Markowitz. Portfolio selection. *Journal of Finance*, 7:77–91, 1952.

[113] Baron Matthew, Brogaard Jonathan, Hagströmer Björn, and Andrei Kirilenko. Return in high frequency trading. *Journal of Financial and Quantitative Analysis*, 82, 2018.

[114] M. Mayo. Evolutionary data selection for enhancing models of intraday forex time series. In *EvoApplications*, volume 7248 of *LNCS*, pages 184–193, 2012.

[115] Luiis Mendes, Pedro Godinho, and Joana Dias. A forex trading system based on a genetic algorithm. *Journal of Heuristics*, 18(4):627–656, 2012.

[116] Andrea Maria Accioly Fonseca Minardi, Antonio Zoratto Sanvicente, and Rogério da Costa Monteiro. Bid-Ask Spreads in a Stock Exchange Without Market Specialists. *Latin American Business Review*, 7(2):19–39, 2006.

[117] Melanie Mitchell. *An Introduction to Genetic Algorithms (Complex Adaptive Systems)*. MIT Press, 1998.

[118] John Moody and Matthew Saffell. Learning to trade via direct reinforcement. *IEEE Transactions on Neural Networks*, 12:875–889, 2001.

[119] Hiroshi Moriyasu, Marvin Wee, and Jing Yu. The role of algorithmic trading in stock liquidity and commonality in electronic limit order markets. *Pacific Basin Finance Journal*, 49:103–128, 2017.

[120] I. V. Morozov and R. R. Fatkhullin. Forex: from simple to complex, 2004. Teletrade Ltd.

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

[122] V. Nekrasov. Kelly Criterion for multivariate portfolios: A model-free approach. *SSRN Electronic Journal*, 10.2139:1–14, 2013.

[123] Steve Nison. Learning Japanese-Style candlestick charting. *Futures Magazine*, December 1989.

[124] Steve Nison. *Japanese candlestick charting techniques: a contemporary guide to the ancient investment techniques of the Far East.* New York Institute of Finance, 1991.

[125] Sebastian Pamela. How program trading works and why it causes controversy in the stock market. *Wall Street Journal*, page 1, 1986.

[126] Robert Pardo. *The evaluation and optimization of trading strategies.* John Wiley, Hoboken, N.J, 2nd ed edition, 2008.

[127] Cheol-Ho Park and Scott H. Irwin. The profitability of technical analysis: A review. *AgMAS Project Research Report*, 2004-4.

[128] L. G. Parratt. *Probability and Experimental Errors in Science.* John Wiley and Sons, 1961.

[129] A. Passamonte. Six facts that give forex traders an edge. *Forex Journal*, 2011. `http://www.forexjournal.com/fx-education/forex-trading/12125-six-facts-that-give-forex-traders-an-edge.html`.

[130] Cristian Pauna. Automated trading software. design and integration in business intelligence systems. *Database Systems Journal*, IX:22–28, 2018.

[131] M. E. H. Pedersen. Portfolio optimization and Monte Carlo simulation. *Hvass Laboratories Report HL-1401*, pages 1–96, 2014.

[132] J. L. Person. *Candlestick and Pivot Point Trading Triggers: Setups for stock, forex, and futures markets.* John Wiley & Sons, 2007.

[133] Riccardo Poli, William B. Langdon, and Nicholas Freitag McPhee. *A Field Guide to Genetic Programming.* Lulu, 2008.

[134] J. Potvin, P. Soriano, and M. Vallee. Generating trading rules on the stock markets with genetic programming. *Computers & Operations Research*, 31:1030–1047, 2004.

[135] William Poundstone. *Fortune's Formula.* Hill and Wang, 2005.

[136] A. Prugel-Bennett. Benefits of a population: Five mechanisms that advantage population-based algorithms. *IEEE Transactions on Evolutionary Computation*, 14(4):500–517, 2010.

[137] UK Reuters Limited, London. *An Introduction to Equity Markets.* John Wiley & Sons, 1999.

[138] R. T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:493–517, 2000.

[139] Boris Schlossberg. *Technical analysis of the currency market: classic techniques for profiting from market swings and trader sentiment*. Wiley trading. John Wiley, 2006.

[140] Bank For International Settlement. Triennial central bank survey of foreign exchange and otc derivatives market activity - preliminary global results, April 2010. `http://www.bis.org/press/p100901.htm`.

[141] J. C. Singleton and J. Wingender. Skewness persistence in common stock returns. *Journal of Financial and Quantitative Analysis*, 21(3):335–341, 1986.

[142] Dong Song, Malcolm I. Heywood, and A. Nur Zincir-Heywood. Training genetic programming on half a million patterns: an example from anomaly detection. *IEEE Transactions on Evolutionary Computation*, 9(3):225–239, 2005.

[143] Hans R. Stoll. The pricing of security dealer services: An empirical study of nasdaq stocks. *The Journal of Finance*, 33, 1978.

[144] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT, 2 edition, 2018.

[145] Michael C. Thomsett. *Bloomberg visual guide to candlestick charting*. Bloomberg visual guide series. Wiley, 2012.

[146] Violeta Todorovic and Aleksandra Pesterac. The impact of automated trading systems on financial market stability. *Facta Universitatis, Series: Economics and Organization*, November:255, 2019.

[147] E. P. K. Tsang and J. Li. EDDIE for financial forecasting. In S. H. Chen, editor, *Genetic Algorithms and Genetic Programming in Computational Finance*, pages 161–174. Kluwer Academic, 2002.

[148] Ruey S. Tsay. *Analysis of financial time series*. Wiley, 2005.

[149] Michael C. Tseng, Soheil Mahmoodzadeh, and Ramazan Gencay. Market impact of algorithmic trading: A reconciliation. *SSRN Electronic Journal*, 2017.

[150] Florin Turcas, Florin Dumiter, Petre Brezeanu, Pavel Farcas, and Sorina Coroiu. Practical aspects of portfolio selection and optimization on the capital market. *Economic Research-Ekonomska Istrazivanja*, 30:14–30, 2017.

[151] Toni Turner. *A beginner's guide to short-term trading*. Jataka tale series. Adams Media, 2002.

[152] A. Vahdat, J. Morgan, A. R. McIntyre, M. I. Heywood, and N. Zincir-Heywood. Evolving GP classifiers for streaming data tasks with concept change and label budgets: A benchmarking study. In A. H. Gandomi, A. H. Alavi, and C. Ryan, editors, *Handbook of Genetic Programming Applications*. Springer, 2015.

[153] G. A. Vasilakis, K. A. Theofilatos, E. F. Georgopoulos, A. Karathanasopoulos, and S. D. Likothanassis. A genetic programming approach for EURUSD exchange rate forecasting and trading. *Computational Economics*, 42(4):415–431, 2012.

[154] Tommi A. Vuorenmaa. The good, the bad, and the ugly of automated high-frequency trading. *Journal of Trading*, 8, 2013.

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

[156] Shimon Whiteson. *Adaptive representations for reinforcement learning*, volume 291 of *Studies in Computational Intelligence*. Springer, 2010.

[157] Bill Williams. *New trading dimensions: how to profit from chaos in stocks, bonds, and commodities*. Wiley trading advantage. Wiley, 1998.

[158] G. Wilson and W. Banzhaf. Interday and intraday stock trading using PAM GP and linear GP. In A. Brabazon, O'Neill, and D. G. Maringer, editors, *Natural Computing in Computational Finance 3*, volume 293 of *SCI*, pages 191–212. Springer, 2010.

[159] Chunchi Wu, Chihwa Kao, and Cheng F. Lee. Time-Series Properties of Financial Series and Implications for Modeling. *Journal of Accounting, Auditing & Finance*, 11:277–303, 1996.

[160] Z. Yin, A. Brabazon, C. O'Sullivan, and P. A. Hamill. A genetic programming approach for delta hedging. *Genetic Programming and Evolvable Machines*, 20(1):67–92, 2018.

[161] Flávio Augusto Ziegelmann, Bruna Borges, and João F. Caldeira. Selection of minimum variance portfolio using intraday data: An empirical comparison among different realized measures for BM&FBovespa Data. *Brazilian Review of Econometrics*, 35, 2014.