TÉCNICO LISBOA

Combining Channel Trading with Genetic Algorithms to Optimize Investments in Trending Forex Markets

Gustavo Marques Nunes Portugal

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor: Prof. Rui Fuentecilla Maia Ferreira Neves

Examination Committee

Chairperson: Prof. Teresa Maria Sá Ferreira Vazão Vasques
Supervisor: Prof. Rui Fuentecilla Maia Ferreira Neves
Member of the Committee: Prof. Rui António Dos Santos Cruz

September 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Gustavo Portugal

To my Parents

Acknowledgments

To my supervisor, Professor Rui Neves, who gave me the opportunity of scratching the surface of the financial markets forecasting thematic, who provided all the guidance I needed to develop this this work, I would like to express my gratitude.

To Tecnico, that was my second home throughout these years and, sometimes, my first home. To all the people I met along the way, specially all the friends I made in AEIST Volleyball Team.

To my manager at Capgemini, who was very understanding and gave me all the flexibility I needed to finish the thesis writing, now I can take all the certifications in the world, António.

But specially, a huge and special thanks to my family, who said many times that I would finish my Master at the age of 40 and would go plant potatoes. Dad, Mom, Bia and Raquel, thank you for all the support you have given me and will always give. I love you all.

Resumo

Este trabalho propõe uma solução inovadora para o comércio no mercado Forex, combinando análise técnica e análise gráfica com Algoritmos de Genética. O padrão de canal e o indicador técnico Fibonacci Pivot Points são responsáveis por classificar o mercado e criar uma estratégia de investimento, que visa prever os movimentos futuros dos preços. Um algoritmo genético é então aplicado para otimizar as regras de negociação. Para treinar e testar o sistema, o par de moedas EUR/USD com dados compostos por amostras de 1 minuto é usado, entre os anos de 2015 e 2020. Na fase de preparação dos dados, os dados ingeridos são reamostrados de acordo com a variação de preço de fecho (ao invés de utilizar um amostragem de tempo fixo), mitigando o problema de sobreamostragem e subamostragem dos mesmos. Os resultados mostram que a solução proposta é lucrativa para vários pares de moedas com características diferentes e reforçam a importância dos Fibonacci Pivot Points na avaliação do risco de uma negociação. Além disso, treinar o sistema com 2 anos de dados EUR/USD (janeiro de 2015 até dezembro de 2016), resultou em 34,9 % de retorno do investimento no período de teste (entre janeiro de 2020 e dezembro de 2020).

Palavras-chave: Algoritmos de Genética, Forex, Padrão de Canais, Análise Técnica

Abstract

This work proposes an innovative solution to trade in the Forex, by combining technical and chart analysis with genetic algorithms. The channel pattern and the technical indicator Fibonacci Pivot Points are responsible for classifying the market and create a trading strategy, that aims to predict future price movements. A genetic algorithm is then applied to optimize the trading rules. To train and test the system, EUR/USD currency pair with 1 minute samples data is used, between the years of 2015 and 2020. In the data preparation phase, the ingested data is resampled according to the close price variation (instead of using a fixed time sampling), mitigating the problem of over and undersampling data. Results show the proposed solution is profitable for several currency pairs with different characteristics and reinforce the importance of Fibonacci Pivot Points in assessing a trade's risk. Moreover, training the system with 2 years of EUR/USD data (January 2015 until December 2016), resulted in 34.9% return on investment in the test period (between January 2020 and December 2020).

Keywords: Genetic Algorithm, Forex, Channel Pattern, Technical Analysis

Contents

List of Tables

List of Figures

Nomenclature

ADX Average Directional Index

ATR Average True Range

B&H Buy-and-Hold

CCI Commodity Channel Index

EA Evolutionary Algorithm

EMA Exponential Moving Average

EMH Efficient Market Hypothesis

EP Evolutionary Programming

ES Evolutionary Strategy

EUR/JPY Currency Pair - Euro/Japanese Yen

EUR/USD Currency Pair - Euro/United States Dollar

Forex Foreign Exchange

FPP Fibonacci Pivot Points

GA Genetic Algorithm

MACD Moving Average Convergence Divergence

MDD Maximum Drawdown

NN Neural Network

NonePP No Pivot Points

NPP Normal Pivot Points

OBV On-Balance Volume

OHLC Open-High-Low-Close

OTC Over-the-counter

ROI Return on Investment

RSI Relative Strength Index

S&H Sell-and-Hold

SMA Simple Moving Average

SVM Support Vector Machine

USD/NOK Currency Pair - United States Dollar/Norwegian Krone

Chapter 1

Introduction

1.1 Overview

The Forex market is the market where currencies are traded, involving always purchasing one currency in exchange for selling another. It is the world's largest and most liquid market where trillions of dollars are traded daily. Forex is an over-the-counter (OTC) market in which prices are quoted by Forex brokers and transactions are negotiated directly with the participants. Those participants include corporations, governments, banks, hedge funds, brokers, investors and speculators, where trades occur 24 hours a day, five days a week in a global decentralized network of markets. Nowadays, due to its low requirements regarding initial capital (resourcing to a mechanism called leverage), Forex has become very enticing to the individual client, who takes a speculative approach trying to benefit from changes in Forex rates in relative short periods of time. However, even for an experienced trader, forecasting price movements may prove to be a difficult task.

The Efficient Market Hypothesis states that markets are informationally efficient, which means their prices already reflect all the information available, thus preventing traders from profiting [1]. Nonetheless, traders challenge this theory by studying and extracting features from historical price data. These features are called technical indicators and are no more than mathematical formulas applied to past price movements, that translate into logical rules aiming to assist a trader assessing market conditions. Technical indicators (derived from technical analysis) may themselves constitute a trading strategy - as is the case of this thesis, in which channel patterns are used to classify the market (i.e., when to buy, sell or to stay idle) and define entry and exit timings – or may be used as features that feed a machine learning algorithm, which in turn is responsible for classification. The term *machine learning* was born in 1952 when Arthur Samuel was developing a computer program for playing checkers. The algorithm that he built eventually evolved to the minimax algorithm, as it is known today. At this point, some works had already been done in this field such as: the Touring Test, developed by Alan Turing in 1950; neurophysiologist Warren McCulloch and mathematician Walter Pitts that proposed the first mathematical model of a neural network. In the decades that followed new algorithms arised and concepts like machine learning, artificial intelligence, data mining, deep learning, among others, started to gain structure and to branch out. Even if these tools were available for a long time, it was only recently that computers gained proper hardware resources that can handle the amount of effort these algorithms take to train, furthermore the lack of financial historical data (available nowadays in real-time) was another obstacle that made the use of machine learning techniques to forecast financial markets price movements infeasible. Although not corroborated by scientific work, a quick search on Google reports that, in 2021, around 80% of the volume of financial trades are algorithmic based.

The aforementioned concepts constitute enough material to approach the problematic of forecasting financial markets. However, besides the metrics that are meant to evaluate the performance of a model, how does one know if it has reached its best performance possible? The present work proposes the use of GA to answer this problem. Genetic algorithms are stochastic search algorithms which act on a population of possible solutions, based on Charles Darwin's theory of natural selection. The potential solutions are encoded as chromosomes. New solutions can be produced by mutating members of the current population, and by mating two solutions together to form a new solution. The better solutions are selected to breed and mutate, the worse ones are discarded. Being an heuristic model, GAs do not return an exact solution, rather they find the best solution in a search-space. By feeding the GA with the parameters that define the forecasting price model, for instance the *kernel function* hyper-parameter of a Support Vector Machine, we are able to fine tune those parameters, thus increasing the forecasting model performance.

1.2 Motivation & Objectives

Many solutions with learning capability applied to financial markets have been proposed so far, yet many methodologies remain unexplored. This work is motivated by the lack of existing solutions for financial markets future price forecasting applying chart patterns trading strategies. This thesis introduces an innovative system for price forecasting, using the channel pattern and technical indicators to generate trading rules and build a classification model. The parameters that regulate the trading strategy are then optimized by a genetic algorithm. The proposed solution is built to trade EUR/USD currency pair in the Forex market.

1.3 Contributions

The main contributions of this thesis are the following:

- Develop a classifier model based on channeling and technical analysis, that provides 3 possible classifications: uptrend, downtrend or sideways.
- Employ GA to improve the classification method.
- Generate positive returns in different environments, i.e. applying the model to currency pairs with completely distinct characteristics regarding volume, volatility and transaction costs.
- Confirm the relevance of Fibonacci Pivot Points in assessing the risk of a trade.

1.4 Thesis Outline

The remainder of this thesis is organized as follows:

- Chapter 2 addresses the theory that supports the work developed, providing the reader the necessary concepts to understand the solution. Also, reviews some of the existing solutions in this thematic.
- Chapter 3 describes the proposed system architecture, explaining in detail its composing modules.
- Chapter 4 evaluates the quality of the solution by using quantitative performance metrics and benchmarking it against other possible solutions.
- Chapter 5 summarizes the work and presents some suggestions for future developments.

Chapter 2

Background and Related Work

2.1 Background

This chapter intends to provide the reader with the necessary concepts to understand the work developed throughout this thesis. It starts by introducing Financial Markets (Section 2.1.1) and more specifically the Forex (Section 2.1.2), since that is the market on which this thesis is focused. Then, the tools used to analyse the market are explored in Financial Indicators Section (Section 2.1.3). Furthermore, Genetic Algorithms (used to optimize this solution) are fully explained (Section 2.1.4) and, finally, some existing solutions on themes addressed in this thesis will be reviewed (Section 2.2), aiming to support some of the decisions taken throughout the development of this work.

2.1.1 Financial Markets

A financial market is anywhere where buyers and sellers trade goods and services (financial assets) in exchange for money. In other words, financial markets are places, that may or not have a physical location, where investors buy intangible assets from those who need funds (sellers), creating liquidity that allow businesses to grow. They work as intermediaries who direct money from investors (or lenders) to sellers (or borrowers), being also characterized by their transparency (in order to ensure that the markets set prices that are efficient and appropriate) and regulation regarding trading, costs and fees.

Financial markets can be divided in 3 main categories: commodities, securities and currency [2]. Commodity market is a marketplace where raw or primary products are traded, such as: oil, gas, coffee, sugar, among others.
A security is normally any financial asset that can be traded, however its definition depends on the jurisdiction on which an asset is being traded. There are 4 different types of securities:

- Equity securities: Commonly referred as stocks, they are financial assets that represent shares of an organization. They differ from the other types of securities by giving ownership of the issuing company, i.e. by becoming a shareholder. The investor may profit by selling his stock for a higher price or by collecting dividends.

- Debt securities: An individual, company or government, issue a debt security that is bought by the lender. Basically, one party lends money to the other with the promise of being repaid plus interest (the interest rate depends on the degree of the risk that buying that security represents). As opposed to equity securities, the lender does not become a shareholder by buying a debt security from an entity. The lender becomes a bondholder.

- Derivative securities: As the name suggests, a derivative is a financial security whose value derives from one or more underlying assets, namely stocks, commodities or other derivatives. It is a contract between two parties that comes in the form of futures, forwards, swaps or options and it is mostly used to hedge risk in the underlying assets.

- Hybrid securities: Complex financial asset that combines the characteristics of two or more types of securities, usually equity and debt securities.

Lastly, the currency market, also know as Forex. Forex is the market where currencies are traded, being the most liquid market in the world, with a daily trading volume of 6.6 trillion dollars, according to the 2019 Triennial Central Bank Survey of Forex and OTC derivatives markets.

2.1.2 Forex

The Forex, also known as Forex, is the largest financial market in the world with a daily trading volume of $6.6 trillion, being open 24 hours per day from Monday to Friday. It is a global and decentralized marketplace where its participants can trade two currencies against each other, meaning that a currency value is always relative to another currency (this is called exchange rate). In Forex, the currencies are always quoted in pairs because while one currency is being bought the other is being sold. These currency pairs are organized in 3 categories: majors, crosses/minors and exotic currency pairs. Majors are the most frequent and subsequently most liquid pairs and they always include USD (U.S. dollar) on one side of the pair, which is present in almost 85% of all transactions made. Crosses or minors correspond to pairs that present high liquidity and include one major currency on one side of the pair except for the USD (for example EUR/GBP). Finally, exotic pairs join one major currency with one from an emergent economy and, in addition to having a much lower liquidity compared to the previously enounced categories, they also tend to be very volatile (which may benefit experient traders but certainly harm new ones).

For example, let's say one wants to trade the EUR/USD currency pair with an exchange rate of 1.2280: the currency to the left of the slash is called the base currency (in this case EUR) and it is the reference for the exchange rate of the traded pair (with a value of one); on the other hand the one to the right of the slash is called counter or quote currency (in this case USD). With an exchange rate of 1.2280, in order to buy 1 EUR it would be necessary to pay 1.2280 USD. On the contrary, if one wishes to sell EUR and buy USD, for each EUR sold, 1.2280 USD are received. Furthermore, in the Forex jargon if a trader wants to buy a certain pair, for instance the EUR/USD, it is called *taking a long position* or *going long*. Otherwise, if a trader is selling a pair then he/she is *taking a short position* or *going short*.

The goal of most Forex's retail level participants is to speculate on currencies (apart from speculators, participants include banks, governments, commercial companies, hedge funds, among others...), that is, for instance, instead of buying a certain currency with the purpose of financing a foreign investment or influence an exchange rate they focus on taking advantage of price fluctuations to generate profit. This thesis proposed strategy also falls into this group. When investing in a currency pair, two prices are given for each forex quote: bid price and ask price. The bid price is the price at which the broker is willing to buy the base currency in exchange for the quote currency, in other words it is the best price at which a trader can sell the currency to the market. On the other hand, the ask price is the price at which a trader can buy a base currency from the broker in exchange for the quote currency (or the price at which a broker is willing to sell to the trader), that means it is the best price at which a trader can buy a currency from the market. As might be implied, the role of the broker is to act as a middleman between a small trader and the market. The difference in the price between the ask and the bid price is called spread and it represents the value charged by the broker in order to intermediate. In addition to the spread, a broker may charge additional fees, for example to provide a software interface that allows the trader to invest. Each trader is responsible for selecting a broker that he/she thinks is most adequate to his/her needs.

Next, concepts that fall within a more technical scope are introduced:

- Pip: Pip is an acronym for *percentage in point* or *price interest point* and it is the unit of measurement used to express the change in value between two currencies. It corresponds to the smallest price change that can occur in an exchange rate [3]. For most currencies (including EUR/USD, which is the one used throughout the development of this thesis) a pip is equivalent to a change of 0.01% in price or the change of one unit in the fourth decimal point. Making use of the example given previously, being the exchange rate of EUR/USD pair 1.2280, if the exchange rate rises to 1.2281 it is said that the rate is now 1 pip higher. The same terminology is applied when discussing bid and ask price. Let's assume for the EUR/USD pair that we have the following set:

$$\begin{cases} askprice = 1.22800 \\ bidprice = 1.22775 \end{cases} \tag{2.1}$$

  The spread, which is given by the difference between ask and bid price (as explained above), would be equal to 2.5 pips.

- Position: According to Hayes [4], a position is defined as the amount of a security, asset or property that is owned by some individual or entity. Opening a position is the act of buying or selling an amount of a security, asset or property, which means that a trade has been established and it lasts until an opposing trade takes place. Closing a position refers to the exact opposite action of opening a position, in other words terminate the trade initially made by selling (buying) a security that was bought (sold) when a position was established. There are two kinds of positions that might be taken: long positions, when a security is bought; short positions, when a security is sold. In financial markets, a trader does not need to own a security to open a short position (sell a security), in an analogy it is like they borrow that amount from a broker to pay them back later (at the moment their position is closed).

- Lot (Position size): When a trader opens a position the amount of units bought or sold are measured in lots. There are predefined lot sizes, being the 4 most common: standard lot (100000 units), mini lot (10000 units), micro lot (1000 units) and nano lot (100 units). In this work, all trades were made with a fixed amount of 1 standard lot, which means that every time a position was opened, 100000 units of a base currency were bought or sold. Trading a standard lot in the EUR/USD pair means that, approximately, every pip the price moves correspond to 10 dollars, the same is to say that the pip value would be $10 per pip.

- Leverage: Despite not being a fundamental concept to comprehend this thesis, a notion of what leverage is will be given because it is an important part of Forex trading. Succinctly, leverage is a mechanism that enables a trader to borrow money from his/her broker in order to invest with more funds than the amount he/she possesses in his/her account [5]. Without leverage, trading a standard lot of USD (base currency) would demand the possession of $100000. Using leverage, the broker only requires that the trader deposits a margin of the total amount invested as a safety measure, preventing the trader's account from going negative. Hypothetically, if a broker requires a 1% margin deposit, with an account of $1000 balance a standard lot could be traded. However, the higher the leverage used the riskier the trade becomes because just as the profits are amplified the losses are too.

Market Chart Analysis - General Concepts

In order to comprehend the behaviour of financial markets, traders often recur to historical price charts. The key tools that enable the reader to understand the chart's analysis made in this thesis are presented below [6][7][8].

- Price Fields:
  - Open: First price that a security was traded during a trading period
  - High: Highest price that a security was traded during a trading period
  - Low: Lowest price that a security was traded during a trading period
  - Close: Last price that a security was traded during a trading period

- Types of Charts: The foundation of technical analysis is the chart. There are three main types of charts used in technical analysis, all of them differing in the way of displaying price data.

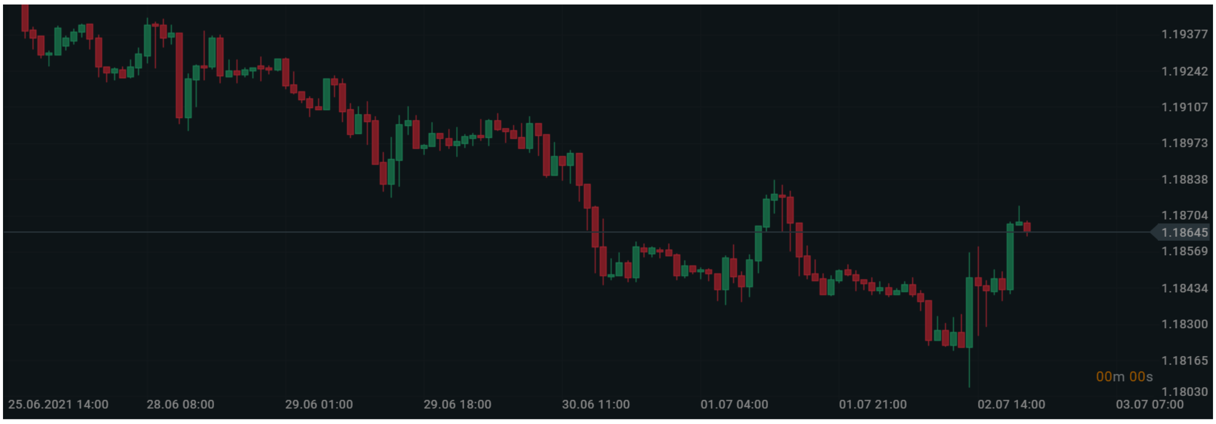
  First, there is the line chart. It is constituted by a single line, usually representing the closing price, where each closing price is linked to the previous closing price creating a continuous line.

  The second option is to use a Bar chart (OHLC chart). Instead of providing only the closing price, bar charts are comprised of a series of vertical lines that indicate the price range during that time frame, showing open, high, low and close prices. The top of each vertical bar represents the highest price that a security was traded during that period, the bottom of the bar represent the lowest price at it was traded, a closing *tick* is displayed on the right side of the bar showing the closing (last) price of that trading period and the opening price is represented by a tick on the left side of the bar. If the chart is colourful, vertical bars that have a closing price superior (inferior) than the opening price are colored green (red), providing a clear view of the market sentiment in that trading period.

Very similar to bar charts, candlestick charts also show open, high, low and close prices. Each trading period is represented by a candle, which is composed by a body and shadows. Bullish candles appear when the market has a tendency of growth (closing price is superior to opening price) and bearish candles appear when the market presents a tendency of decline (where closing price is inferior to opening price). The wider part of the candlestick is called body and provides open and close prices of that period, following the same color schema as the bar chart (if the chart is colourless, bullish candles are white and bearish candles are black). The thinner parts of the candlestick are called shadows and display the highest and lowest price of a certain trading period. Figure 2.1 shows an example of the different types of charts referenced above.

(a) Line chart

(b) Bar chart



(c) Candlestick chart

Figure 2.1: Chart representations for EUR/USD currency pair.

- Trend: A trend can be defined as the phenomenon by which price movement tends to persist in one direction for an extended period of time. Market moves are made in a series of *zigzags*, resembling a series of successive waves with peaks and troughs, being the direction of these peaks and troughs what constitute a trend. As Figure 2.2 shows, if there is a successive series of higher troughs for a period of time, then there is an uptrend. On the other hand, if a successive series of lower peaks is verified, then there is a downtrend. If none of these patterns is verified, then the market is sideways.

Figure 2.2: Types of trends (adapted from https://capital.com/4-tips-on-how-to-spot-a-market-trend-before-it-gets-obvious

- Support and Resistance: The peaks and troughs that determine the trend of the market are called, respectively, support and resistance. In a chart, support is defined as the zone where the buying interest is sufficiently strong to overcome selling pressure, therefore when a security's price drops it tend to hit the support zone and bounce back (price increase). Resistance is the opposite of support and is defined as the area where selling pressure overcomes buying pressure. Again, when the price reaches a resistance area, it tend to drop again. Achelis [6] offers an intuitively definition of support and resistance and its relation with the principle of supply and demand. Once a support (resistance) is penetrated, it becomes the new resistance (support), as can be seen in Figure 2.3 where Resistance 1 becomes Support 4 in an uptrend.

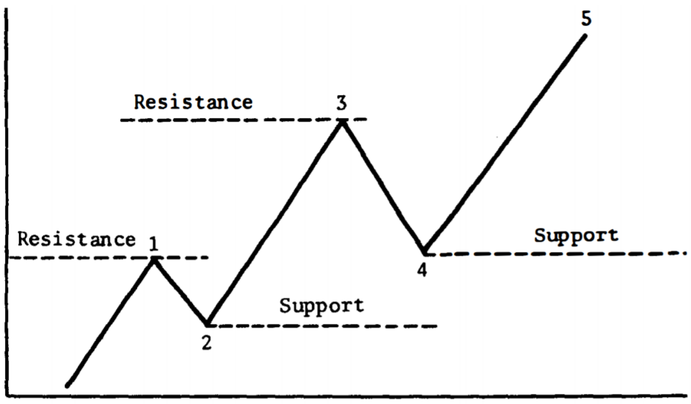


Figure 2.3: Support and resistance levels in an uptrend (adapted from *TECHNICAL ANALYSIS OF THE FINANCIAL MARKETS* [7]).

2.1.3 Financial Indicators

In financial markets, the success of an investor relies on the quality of the information used to assist in the decision-making process and on the velocity at which those decisions are made [1]. Two common approaches used to analyse price evolution along time are fundamental analysis and technical analysis. Fundamental analysis acts on the belief that the market does not reflect the true intrinsic value of a security at any given moment, hence analysts make predictions based on the principle that those securities must be undervalued or overvalued. To generate those predictions, fundamental analysts study the factors that may affect a security's value – from macroeconomic and industry conditions to micro economic factors, like the company's management effectiveness [9] - thus yielding indicators of future price movements. On the contrary, technical analysts consider that all the necessary information to analyse the market is already reflected in the price of a security [10]. Having said that, they support their investment decisions based on past price movements, since they assume the existence of patterns in price that repeat themselves and hence can be anticipated. Although these two approaches (fundamental and technical analysis) are normally used independently, they may be applied together, as Bettman, Sault and Schultz show in their study [11]. Notwithstanding any of the above referred approaches could have been used separately or in a hybrid model, in this thesis it was decided that the system developed would only take technical analysis into consideration, thus, and as the name suggests, only technical indicators are employed.

The major objectors to fundamental and technical analysis are the *Random Walk Theory* [12] and the *Efficient Market Hypothesis* [1]. According to the Random Walk Theory, if the flow of information is unimpeded and information is immediately reflected in stock prices, then tomorrow's price change will reflect only tomorrow's news and will be independent of the price change today, meaning that past movements in stock prices cannot be used to predict its future movements. Associated with this theory is the Efficient Market Hypothesis, which states that stock markets reflect all the information available, therefore stocks are traded at their fair value. As a consequence, when new information arises it is almost immediately incorporated in price, making it infeasible for traders to profit [13]. Nevertheless, several investors, financial economists and brokerage firms have used technical analysis in stock market prediction with considerable success. [1]. Although a consensus has not been reached regarding the aforementioned thematic, evidence suggests that markets are to a certain extent predictable [14], thus validating the use of fundamental and technical analysis to examine the market behaviour.

Technical Analysis

As opposed to fundamental analysis, technical analysis does not utilize any external economic data or any relevant news events, neither the characteristics of a company or industry [15][1].As mentioned before, technical analysis produces trading rules exclusively based on past price movements and it is based on three premises[7]:

- Market action discounts everything: Technicians believe that all the fundamentals that influence the market movements are immediately incorporated in the price.

- Price moves in trends: Predictable trends are essential to the success of technical analysis. When spotted early, one safe way traders may profit is by investing in the same direction as the trend. As technicians counsel, “the trend is your friend”[16].

- History repeats itself: Asset traders will tend to react the same way when confronted with the same conditions.

Based on these premises, technical analysts are able to abstract the fundamentals of a company. Succinctly, technicians try to identify patterns in price, volume, breadth and trading activities, that enable them to spot trends and reversal of trends.

A technical indicator may be described as a series of data points derived from a mathematical formula that is applied to the price data series, reducing the price prediction problem to a pattern recognition problem where the inputs are the historical prices[17]. Price data includes any composition of the opening, high, low or closing values over a period of time as well as volume data[18].

According to Yong Hu’s et al. research [19], technical indicators may be categorized into eight different groups where each group provides different information about a security’s value, complementing each when used in conjunction in order to yield buy and sell signals . The ones that are considered the most used and relevant in FX trading are described below, the remaining (sentiment, flow-of-funds, raw data, cycle) may be consulted in [19]:

- Trend: They are a type of price-based indicator for tracing the stock price trends. Indicate the strength and direction of the price move. Common trend indicators include SMA, EMA, MACD, ADX, among others.

- Momentum: Price-based indicators, used to evaluate the velocity of price change and spot trend reversals. Tend to work better on bigger time frames (daily or weekly), since these can be erratic on short frames. Common momentum indicators include RSI, CCI, Stochastic Oscillator, among others.

- Volatility: Evaluate the deviation of a security’s price from its mean value, which reflects the fluctuation range of historical prices. It can be used to evaluate the risk and identify the level of the support and resistance. Prices are generally recognized to fluctuate between the level of support and resistance, but continue to rise (fall) once they break through the level of resistance (support). Common volatility indicators include average true range and Bollinger Bands, ATR, among others.

- Volume: Reflect the enthusiasm for investing of both buyers and sellers, which is also a basis for predicting stock price movements. Strategy applying volume indicators is grounded on the hypothesis that price movement is determined by the enthusiasm of buyers and sellers. Trends are viewed as more sustainable when volume is rising. Common volume indicators include OBV, Chaikin Money Flow, among others.

Regardless of its category, technical indicators are split in two major groups depending on the timing for which they give information:

- Leading indicators: Attempt to forecast future price movements using past price data. This kind of indicators enable a trader to open a position right in the beginning of a trend, hence increasing the amount of profit, however these are prone to false signals (i.e., indicator gives a signal while the price moves in the opposite direction).

- Lagging indicators: Price movements occurs before the indicator signals it. These are used to confirm a trend and, despite of forcing a trader to lose the beginning of a price movement, their risk of returning false signals is significantly lower when compared to leading indicators.

When creating a trading strategy, a combination of leading and lagging indicators, only leading, only lagging or just the study of price charts themselves are valid possible approaches.

For further explanation concerning specific technical indicators [6][7][8] should be consulted. In the remainder of 2.1.3, the technical indicators applied in this work and their motivation are explained.

- Zig Zag: It is a technical indicator used to find price trends while eliminating insignificant price movements that occur within the overall trend (also called market noise). It does it by selecting peaks (swing highs) and troughs (swing lows) where price movements between a swing high and a swing low exceeds a pre-chosen value, usually 5% [20], as exemplified in Figure 2.4 . The algorithm is as follows:

  1. Choose a starting point (swing high or swing low).
  2. Choose a price movement threshold (in percentage or in pips), for instance let's say this value corresponds to 5%.
  3. Find next swing high or swing low whose price differs more than 5% from the starting point, note that after a swing high (low) must come a swing low (high).
  4. Draw a trend line from the starting point to this new point.
  5. Repeat this process, considering that the new point generated becomes the starting point.

  Zig Zag is a lagging indicator, therefore it cannot be used to predict future market movements. Instead, in this thesis trading strategy, it is used to: smooth price charts, find trends and finally, having identified a trend, find possible entry points (i.e. the price at which a trader opens a position).

Figure 2.4: Zig Zag indicator (adapted from [20]).

- Fibonacci Pivot Points: In order to define Fibonacci Pivot Points, we first need to introduce the concept of Pivot Points and the role that Fibonacci numbers play in this indicator.

  Pivot Points are a leading technical indicator used to predict market trends and, most important since it is their purpose in this work, to find areas of support and resistance. By anticipating support and resistance zones, this thesis solution uses pivot points as an entry signal that alerts if we are in the conditions to open a position [21].

  Leonardo Fibonacci, the well known mathematician, created a sequence of number, starting with 0 and 1, where a number is found by adding up two numbers before it. The sequence goes 0, 1, 2, 3, 5, 8, 13, 21, 34 and so forth until infinity.The most important contribute of Fibonacci sequence is the so called *golden ratio*, which is result of dividing one number of the sequence by the previous one (starting on 5) and it is equal to 1.618 or its inverse, that is equal to 0.618. The *golden ratio* has been verified in the most diverse areas of nature, such as: the ratio between our forearm and hand is 1.618, the number of female bees in hive divided by the number of males is also equal to 1.618 and so on. Financial markets do not constitute an exception since many traders believe this relation affects price movements. Whether it is true or not, if the majority of traders act according to this rule, they will inevitably affect the market. When used in technical analysis, the *golden ratio* normally translates into three ratios: 0.382, 0.50, 0.618.

  That being said, Fibonacci Pivot Points are Pivot Points that incorporate Fibonacci ratios in their calculation. There are seven basic Fibonacci Pivot Points and they are calculated using the high, low and close prices of the previous trading period:

  - Base Pivot Point: $P = (High + Low + Close)/3$

– Support 1: $S1 = P - (0.382 * (High - Low))$

– Support 2: $S2 = P - (0.618 * (High - Low))$

– Support 3: $S3 = P - (High - Low)$

– Resistance 1: $R1 = P + (0.382 * (High - Low))$

– Resistance 2: $R2 = P + (0.618 * (High - Low))$

– Resistance 3: $R3 = P + (High - Low)$

If the price is above the Base Pivot Point then the market is bullish, otherwise it is bearish. An example of the application of Fibonacci Pivot Points is provided in Figure 2.5.

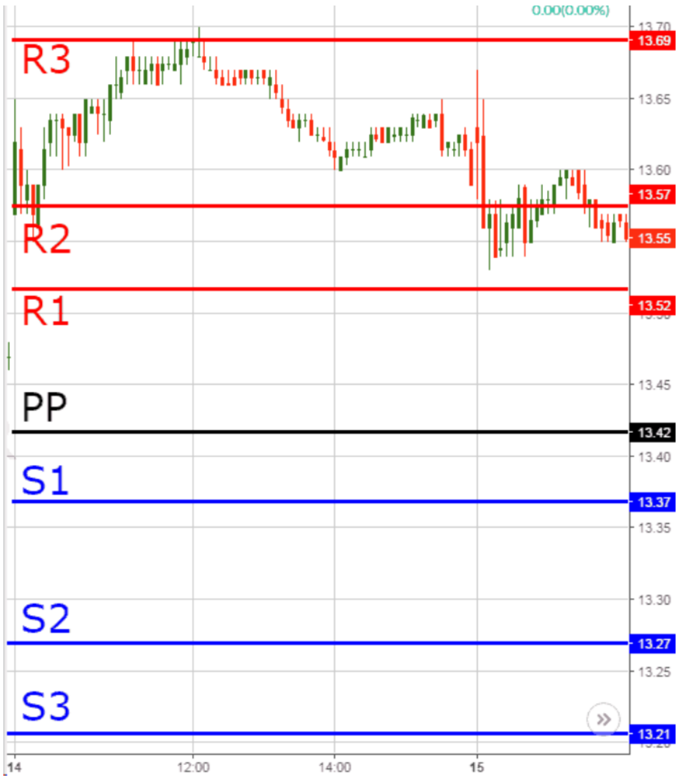


Figure 2.5: Example of a bullish market with its respective Fibonacci Pivot Points drawn (adapted from https://tradingsim.com/blog/pivot-points/Fibonacci_pivot_points_are_most_popular).

2.1.4 Genetic Algorithms

Evolutionary algorithm (EA) is an umbrella term used to describe population-based stochastic direct search algorithms that in some sense mimic natural evolution [22]. EAs use heuristic methods, which means that, unlike traditional methods, they don't return an exact solution. Instead, they give the best approximate solution. In conclusion, they trade precision for computer resources optimization. There are four kinds of EAs: Evolutionary Programming (EP), Genetic Algorithms (GAs), Evolution Strategies (ESs) and Genetic Programming (GP). All of them possess the same characteristics: points in the search space are considered as individuals (forming a population), the quality of a solution is evaluated by a numerical fitness function, use initialization and termination criteria, apply the concept of operators and their representation is given by a genotype-phenotype mapping.

The genetic algorithm is a search heuristic model applied to optimizations problems, based on Charles Darwin's theory of natural selection. The principle behind this algorithm is that by picking an initial population and applying the principle of natural selection (where only the fittest individuals survive) for a finite period of generations, the output will converge to the best possible solution, which means a global maximum (from a search problem perspective) or the best fitted individuals (from a biological perspective).

A genetic algorithm is composed by six phases, drawn in the flowchart of Figure 2.6:

1. Initial population
2. Fitness function
3. Selection
4. Crossover
5. Mutation
6. Termination Criteria

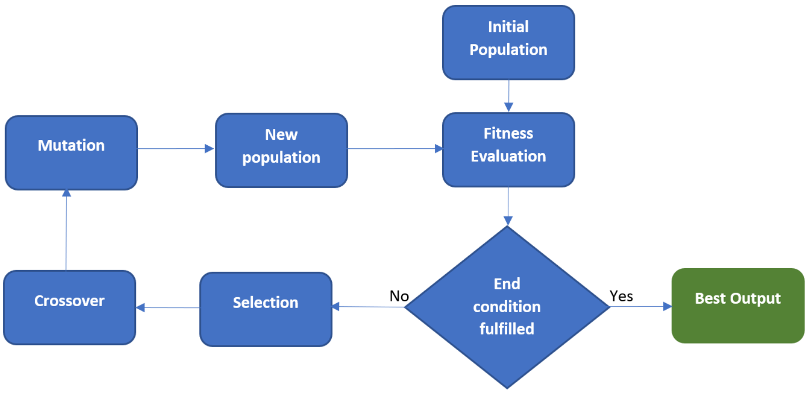


Figure 2.6: Generic flowchart of a GA's phases.

Initial Population

The algorithm starts by creating an initial population. The population is composed by individuals, where each individual represents a possible solution to the problem, therefore the number of possible solutions is equal to the population size. Each individual has a chromosome which, in turn, is formed by a set of genes. Each gene corresponds to a specific feature we want to optimize with the GA. Chromosomes may be encoded in different ways, such as: binary (each chromosome is represented as a string of zeros and ones), permutation (each chromosome is represented by a string of numbers that represents a position in a sequence), value (each chromosome is represented by a string of its actual value, may be a real or integer number, for instance). Figure 2.7 shows the structure of the population.

Evaluation

Each member of the population is evaluated according to a fitness function. The fitness function determines how fit an individual is in fulfilling the problem requirements. The higher the fitness score, the higher the quality of the solution and the greater the probability for a chromosome to be selected for reproduction.

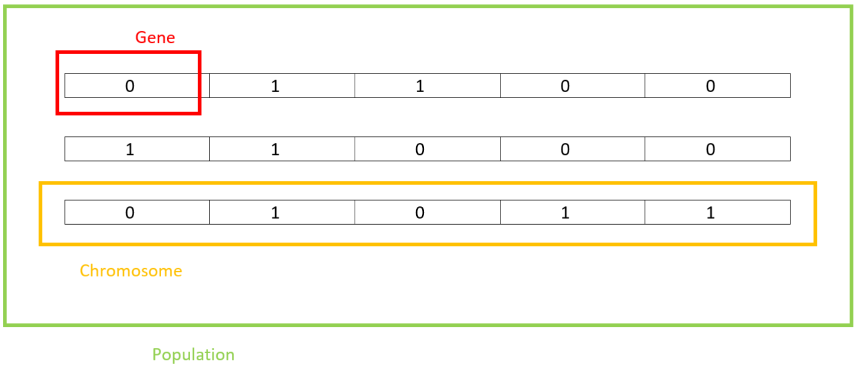


Figure 2.7: Structure of a population's individuals.

Population size affects the quality of the final solution. A greater amount of individuals increase diversity in the population, hence the chance of obtaining a better solution also increases. However, as the population size increases, the algorithm takes more time to converge. On the other hand, if the population is composed by a reduced number of individuals, it is more likely the algorithm will not return an accurate solution. In conclusion, an equilibrium needs to be established between performance and accuracy.

Selection

The selection phase determines which individuals are chosen for mating (reproduction) and how many offspring each selected individual produces. The idea is that by selecting the individuals with the best fitness score of a population and recombine their genetic material (crossover and mutation), they will give origin to even more adapted (higher fitness score) individuals. Even though the best fitted individuals are the ones with a best chance of improving the next generation, worse fitted individuals also need to have a chance of participating in the reproduction process, otherwise there is a risk of reducing the variety of genetic material too much and the algorithm may not converge to optimal solutions. The variable that regulates the degree to which better individuals are favored at the expense of the worse fitted is called selection pressure.

The techniques used in this thesis and others, are addressed in more detail in [23][24][25]. The different ways of doing parent selection fall into one of the following categories:

- Fitness Proportionate Selection: The probability of an individual becoming a parent is proportional to its fitness. Therefore, better fitted individuals have a higher chance of propagating their genes to the next generations. There is selection pressure involved.

- Tournament Selection: Select K individuals from the population at random and the best out of this becomes a parent.There is selection pressure involved.
- Random Selection: Parents are randomly selected from the population. No selection pressure involved.
- Rank Selection: A rank is given to each individual according to its fitness score. Then, whichever individual gets a pre-chosen rank number gets selected as a parent. In a way, it is the same as random selection. No selection pressure involved.
- Elitism: Best fitted individuals are granted a place in the next generation. No selection pressure involved.

Elitism or Elitist selection is a way of granting that the best fitted individuals propagate their genetic material to the next generation. While it may compromise the amount of diversity (of genetic material) which leads to premature convergence, if used appropriately improves the quality of the solution. To overcome this problem, in this work, elitism is responsible for generating only 10% of the next generation, the remaining 90% are selected using tournament selection. There are some variations of elitist selection: in its most simple form, the selected individuals are passed directly to the next generation based on their fitness score; another way would be to perform another type of selection in the entire population and then, *x%* of the worst fitted individuals would be replaced by the *x%* best fitted of the current generation; other strategies can be applied, depending on the algorithm needs.

In tournament selection, explained by figure 2.8, *k* individuals from the population are randomly selected. Then, they compete against each other winning the individual with highest fitness. This individual is selected to reproduce. With this strategy, all individuals get a chance of being selected as parents providing a way of preserving diversity in the population. The parameter *k* is called tournament size and, as *k*, the chances of a less fit individual be selected to reproduce decrease, which increases selection pressure and may lead to lack of diversity in genetic material.

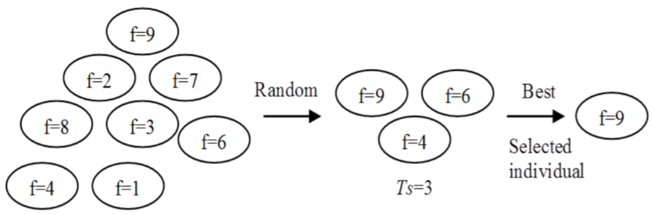


Figure 2.8: Tournament selection procedure (adapted from [24].

Crossover

The next phase is crossover. Once the parents are selected, every two parents are recombined in a process that mimics natural reproduction, yielding offspring. This recombination is made by swapping genes between two parents and there is a considerable amount of crossover techniques that can be employed. A.J. Umbarkar and P.D. Sheth presented a study with some of them [26]. Some of the most used crossover techniques are one-point crossover, multi-point crossover and uniform crossover.

In one-point crossover, a crossover point is chosen at random and a offspring is created by exchanging the genes of parents among themselves, starting on the crossover point, as Figure 2.9 shows.

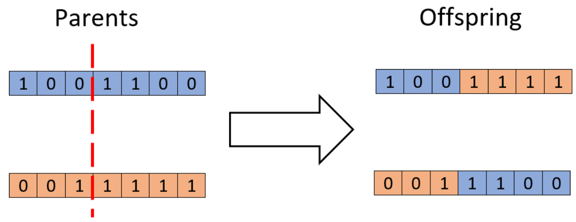


Figure 2.9: One-point crossover.

Multi-point crossover works exactly like one-point crossover, excepting for the fact that the number of crossover points is bigger than one, as Figure 2.10.

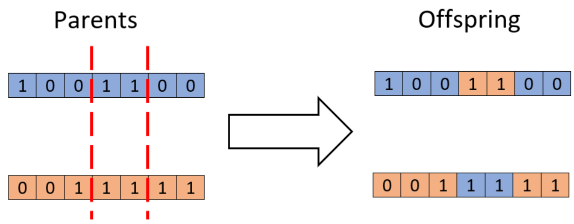


Figure 2.10: Multi-point crossover.

In uniform crossover, each gene is treated separately. For each position in the chromosome, the parent who contributes with its gene is chosen randomly with equal probability, just like flipping a coin, as Figure 2.11 shows. However, this parameter can be set to benefit one parent more than the other, for instance: parent 1 genes have a probability of 30% of being chosen and parent 2 a probability of 70%. Uniform crossover starts by swapping large amounts of genetic material, like one-point crossover.

So, initially the two operators may behave similarly, and uniform crossover is initially a global search operator. As the population starts converging, uniform crossover becomes more and more local in the sense that, the offspring it produces are progressively more similar to their parents. However, unlike multi-point and one-point crossover, at this stage the search is, in some sense, largely unbiased as any node in the parents has the same chance of being inherited by the offspring [27].

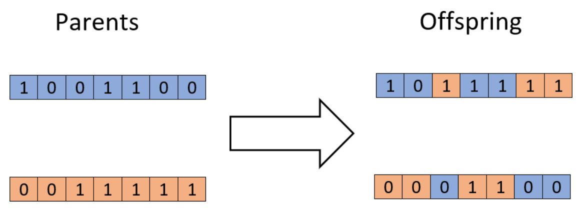


Figure 2.11: Uniform crossover.

Mutation

Mutation is an operator used to introduce divergence in the population. Without it, the offspring would contain only the characteristics of its parents, increasing the risk of premature convergence leading to a non global solution. To apply mutation: choose a mutation rate (for example 5%), then all genes of the chromosome (this can be set to only mutate some specific genes) have a 5% probability of being swapped with a random value that exists in the search space. For instance, if a gene is represented by an integer value that may vary between 1 and 10 and, if this gene contains the value 3 before mutation occurs, there is a 5% chance of this gene changes to a random value between 1 and 10.

Termination

The algorithm ends when the population has converged. In practice, a termination criteria may be defined by: establishing a limit number of generations (for example, it is assumed that after 200 generations the algorithm has converged to an optimal solution) or by defining a consecutive number of generations where there is no evidence of evolution from the population.

2.2 Related Work

This chapter presents some of the most relevant works found in the literature in the context of this thesis, which is to build a trading strategy for the FX market using technical analysis and optimizing the strategy parameters with a GA. Moreover, they provided a baseline and answered some simple questions that were raised throughout the development of this work: Should one use technical or fundamental analysis to study the FX market? What are the important considerations to keep in mind when designing a channel chart pattern? How is an automated trading system built? How does a GA work and which variations may be incorporated? Somehow, all the works referenced in this Section played a part in answering these questions. Certainly, they do not cover all the different possibilities that could have been applied in this trading system neither they explain all the decisions made for this particular solution but, as stated before, they provide a solid starting point.

2.2.1 Forex and Technical Analysis

It is not common to trade by inspecting channel pattern formations and, throughout the entire research, only one study was found regarding this thematic. Dempster and Jones [28] used the channel pattern trading strategy to forecast price movements in the FX market, namely they apply the strategy to the GBP/USD currency pair. In order to form a channel, 4 points (2 peaks and 2 troughs) are selected using a considerable complex algorithm described in the paper. Having a channel, the entry point to trade is when the price has moved 2% of the vertical channel width in the direction of the channel. A trade is exited (position closed) if: the market moves $y\% \times channelwidth$ or 3 pips in the opposite direction of the channel, if the price hits the target wall or if the market moves $15\% \times channelwidth$ from the source wall outside of the channel. Without the assistance of any intelligent algorithm to refine all the variables involved, their strategy proved to be generally unprofitable, however they were able to find evidence of a correlation between the channel pattern and profitability, that is not accounted for in the Efficient Markets Hypothesis. Also, they were able to prove that the profit largely depended on the channel width. At a primordial phase of the development of this thesis, a similar way of capturing channel patterns (using the same algorithm as in this study to create the channel but applying different trading rules) was tested and the results showed that only a fraction of channel formations that resulted in up or down trends were being captured. A second attempt made, increasing channel width, ended up capturing a lot of sideways markets. In 1998, Lui and Mole [29] reported the results of a questionnaire survey, conducted in February 1995, on the use of fundamental and technical analysis by Forex dealers in Hong Kong, to forecast exchange rate movements. Their findings reveal that 85% of respondents rely on both fundamental and technical analyses for predicting future rate movements at different time horizons. At shorter time horizons, there exists a skew towards reliance on technical analysis as opposed to fundamental analysis, but the skew becomes steadily reversed as the length of the horizon considered is extended. The most common length of historical data used by the dealers is 12 months and the used trading period is daily data. Technical analysis is considered slightly more useful in forecasting trends than fundamental analysis, but significantly more useful in predicting turning points. Interest rate related news is found to be a relatively important fundamental factor in exchange rate forecasting, while moving averages and other trend-following systems are the most used techniques. In 2018, Jakpar et al. [10] also compared technical and fundamental analysis. They conducted a study to analyse the credibility of both, the fundamental and technical analysis, on predicting the Malaysian stock market return. Their

study included 80 companies from the food market industry between the years of 2012 to 2016. The fundamental indicators included net profit margin, price earnings ratio and total asset turnover while for technical analysis only MACD was considered. Although both approaches showed predictive ability and positive returns, fundamental analysis proved to be slightly more accurate. The author justifies this result with the fact the study is focused on a specific industry and the number of indicators used for each approach is different (3 for fundamental analysis, 1 for technical), referencing three other studies where technical analysis outperformed fundamental analysis - Needly (2010), Moosa and Li [30] (2011) and also Wafi et al. [31] (2015).
In 2010, Teixeira and Oliveira [17] presented a trading strategy for stock trading, tested on 15 companies quoted in São Paulo Stock Exchange, combining technical analysis with the nearest neighbour classification (k-NN algorithm). Their goal was to study the feasibility of using an intelligent prediction system exclusively based on the history of daily stock closing prices and volumes. To build a daily trading model, the authors employed a diverse set of technical indicators that fed the model (such as: RSI, SMA, Stochastic Oscillator, Bollinger Bands, among others; and all of them with standard parameters setup), a stop loss of 3% of drop in the closing price and a stop gain of 10% of rise in the closing price. They achieved a classifier precision of around 40%, with an average accumulated profit of around 500% in the period between 1998 and 2009, depending on slightly different variations of the algorithm. Compared to the B&H, this strategy outperformed it by a minimum of 34.50% and a maximum of 58.10%, regarding accumulated profits. In 2016, Coakley et al. [32] went further in the research of profitability with technical trading, as they conducted an investigation of profitability of FX technical trading rules, over the period between 1996 and 2015, for 22 currencies quoted in US dollars. This translated in a total of 113448 trading rules using essentially the most common technical indicators in order to build those rules, such as: MACD, RSI, MA's, Bollinger Bands, etc. Their findings suggest that prior to controlling for data snooping bias (data snooping bias happens when a person, intentionally or not, refines too many parameters to increasing a system's performance on a single data set, influencing the final results), quite large numbers of technical trading rules are significantly profitable and can achieve annualised returns up to 30%. Regarding trading systems, et al. [1] performed a study that aimed to analyse price forecasting, using a broad set of machine learning models applied to several financial markets.

2.2.2 Genetic Algorithms

In order to optimize the proposed system's trading strategy, a GA was used. [22][19][33] constitute a good starting point to understand how an evolutionary algorithm (EA) works and explore some variations of operators and their applications. Given the theme of this thesis, most of the referenced works study propose different approaches for creating trading systems in the Forex optimized by a genetic algorithm.

In 2004, Elshamli et al. [34] introduced a genetic algorithm approach for solving mobile robot path planning problems in static and dynamic environments. The Genetic Algorithm Planner (GAP) utilized variable length chromosomes for path encoding, where the population represents the path and each gene of the chromosome is represented by (x,y) coordinates of each point. In order to evaluate the fitness of the individual they used the technical indicator standard deviation. The experiments included 4 different approaches: the first using low convergence rate, in an attempt to increase diversity (i.e. set a high mutation rate and a low crossover rate at the same time); using memory, the GA stores the best solutions so far and keeps replacing the worst individual with these solution at a prefixed rate; using random immigrants, by replacing the worst fitted individuals with randomly generated ones; finally, a combination of memory and random immigrants. Although all solutions produced good results, the combination of memory and random immigrants is the best among the four. In the same year, Evans et al. [35] developed an intra-day FX trading system, using a hybrid solution that combines NN and GA. The forecasting model was based on Feed Forward Neural Networks with BackPropagation fed with data from the three most traded major currency pairs (EUR, GBP, EUR), using a GA to optimize the network topology. In the selection process they used an elitism policy of just one individual (elitism ensure the best genetic material is kept throughout generations) and roulette-wheel selection for the remaining population. The rest of the operators used the following configuration: a two-point crossover was applied with 80% of crossover rate, mutation rate of 20% and a termination criterion of 40 generations or 15 generations without evolution. Their work confirmed with a significance of 95%, that daily Forex currency rates time series are not randomly distributed, obtaining 23.3% of annualized net return and 72.5% of prediction accuracy. The evaluation function was composed by a combination of the mean absolute error, sharpe ratio and annual net return. In the work they developed for portfolio optimization, Aranha and Iba [36] also proposes a multi-objective fitness function, where cumulative return evaluated the solution in terms of profit generated and sharpe ratio evaluated the risk associated to an investment. In 2009, Hirabayashi et al. [37], proposed a Genetic Algorithm (GA) system to automatically generate trading rules based on Technical Indexes, focusing on calculating the most appropriate trade timing, instead of predicting the trading prices. The features that feed the GA were derived from known technical indicators, for instance: one trading rule, named PD, gives the percent difference between the simple moving average (SMA) and the actual price. The strategy took leverage into account, where each investment permits a maximum leverage of 10 times. Concerning the GA specifications, the individuals containing the trading rules features were binary represented and profit was used to evaluate the fitness of the population. Furthermore, the parents were chosen by tournament selection - a method later studied by Shukla et al. [23] and Razali and Geraghty [24], where they came to the conclusion that it outperformed other selection methods in terms of convergence rate and time complexity - having a tournament size equal to 50, which is not a huge number having into account the population is composed by 1500 individuals (corresponding to 3.33%). In order to make sure that good genetic material was preserved, elitism was performed passing 1% of the best fitted individuals to the next generation. Also, the worst 30% fit individuals were replaced by random immigrants. Finally, a two-point crossover and mutation were performed, with rates of 60% and 1%, respectively. To train and test the algorithm, 3 currency pairs were taken into account with a fixed spread each - USD/JPY, EUR/JPY, AUD/JPY – training and testing for a period of 4 years. To separate the training and test data, the authors used a rolling window method, which means they trained for a 6-month period, and tested for a 3-month period. Then, for each subsequent experiment, they moved both the training and the testing period 3 months forward. During the 4 years period, the results showed a maximum profit gained of: 80% for EUR/JPY, 17% for USD/JPY and 38% for AUD/JPY. Also, a comparison between the proposed solution and the same solution without leverage, NN and B&H strategy revealed that the first outperformed the remaining during the entire period tested. In 2010, Myszkowski and Bicz [38], developed a study that is worth mentioning for its particular combination between decision trees and a genetic algorithm and its distinct selection method. That being said, they presented a genetic algorithm application to generate profitable strategies to trade futures contracts on Forex. Strategy model was based on two decision trees, responsible for taking the decisions of opening long or short positions on EUR/USD currency pair. In turn, trees took into consideration only technical analysis indicators: SMA, WMA, DMA, RSI, MACD, ROC, volume and volatility. Each individual was represented as two decision trees, where one is responsible for buy signals, the other for sell signals. The authors used an innovative selection method by combining tournament selection and roulette wheel selection: with a tournament size of 5, 5 individuals were randomly selected; then, instead of picking the best individual to reproduce, these were sorted and a probability of being chosen is given according to its fitness. Zhang and Ren [39] focused on making 5 minutes predictions for the EUR/USD currency pair. Aiming to find the best combination of entry and exit rules, they recurred to a GA, with the particularity of having as fitness function the technical indicator sterling ratio, not commonly found in the literature. In 2013, Deng and Sakurai [40] studied the feasibility of using only the Relative Strength Index (RSI) technical indicator to predict FX price movements, evaluating the indicator results for multiple timeframes at the same time. The target trading currency pair was EUR/USD and trading time horizon one hour. They used a GA whose goal was to search for the best parameters of the RSI indicator. To evaluate the results the authors computed the ROI, standard deviation and benchmarked their solution against buy-and-hold (B&H), sell-and-hold (S&H), SVM classification and a GA-RSI solution targeting only one timeframe. The proposed solution (GA-RSI-multiple timeframe) outperformed the benchmarked solutions, achieving an average 13.87% of ROI per month, concluding that using more than one timeframe can improve the assessment of overbought and oversold conditions. In 2020, Zhang and Khushi [41] proposed an automatic robotic trading strategy using 16 technical indicators to construct trading rules. Then, a GA optimized the weight of each technical indicator in the decision-making process, in order to maximize sharpe ratio and sterling ratio. The combination of these two metrics constitutes the fitness function of the GA. To test the system's performance, an experiment, on intraday data, data frequency of 5 minutes per sample, of 6 major currency pairs from 2018 to 2019, was conducted. The results showed that their strategy was able to achieve an annual return of 320% with a 1:20 leverage in AUD/USD currency pair. However, the proposed strategy did not consider transaction fees. Furthermore, this strategy outperformed B&H, S&H and GA-MR, which is the same strategy but maximizes the return, instead of maximizing sharpe ratio and sterling ratio.

2.3 Chapter Conclusions

In the first part of this chapter, financial markets are introduced. Then, the Forex and the tools necessary to understand the trading strategy developed in this work are provided. It is followed by an explanation about genetic algorithms (which optimize the proposed strategy) and its main phases of implementation. Lastly, relevant works found in the literature that contributed to this thesis are presented, being important to leave one note: despite the acknowledgement of the existence of other features that may enrich a genetic algorithm, such as hyper-selection and hyper-mutation (which increase selection pressure and mutation rate, respectively, whenever there is a degradation in the performance of the GA), those approaches were discarded in this solution, taking the risk of adding too many variables to control in the algorithm. Among all the works referenced, table 2.1 summarizes the most relevant ones.

Work	Date	Methodologies	Approach	Evaluation Function	Result Function	Financial Application	Period	Best Return
[28]	2010	TA	Channel Pattern	Profit Gained	% correct, % better than chance	FOREX – GBP/USD	1989-1997	13 pips for down-channels
[17]	2010	NN, TA	Stop Loss, Stop Gain, RSI filter	Accumulated profit	Comparison with B&H	FOREX – GBP/USD	1998-2009	511.35%
[34]	2004	GA	LC, RI, M, MRI	Best solution	Comparison of the 4 variations	---	---	MRI
[35]	2004	GA, NN, TA	Elitism, Roulette-Wheel Selection, Termination Criteria of 15 generations unchanged	Mean absolute error, Sharpe Ratio, Annualized Return, Correlation Coefficient	Monte Carlo Simulation, Random Method, comparison between actual data and predicted data	FOREX – EUR/USD, GBP/USD, EUR/GBP	2010-2012	23.3% (Annualized Return), 72.5% (Prediction Accuracy)
[40]	2013	GA, TA	Several timeframes analysis at the same time	ROI, Standard Deviation, A/S ratio	Comparison with B&H, S&H, SVM, GA-RSI-single-timeframe	FOREX – EUR/USD	2011	13.87% (ROI per month)
[37]	2009	GA, TA	Tournament Selection, Elitism, Random Immigrants, Leverage	Profit Gained	Comparison with solution without leverage, NN, B&H	FOREX – EUR/JPY, USD/JPY, AUD/JPY	2005-2008	80% EUR/JPY, 17% USD/JPY, 38% AUD/JPY (ROI in 4years)
[39]	2010	GA, TA	Roulette Wheel Selection, Sterling Ratio, neutral position (instead of only buy and sell)	Sterling Ratio, ROI	Comparison neutral vs. without neutral	FOREX – EUR/USD	2009	3.7%
[38]	2010	GA, TA, Decision Trees	Hybrid Selection – Tournament Selection combined with Roulette Wheel Selection	Profit Gained, Avg. Profit, Avg. Profit incl. prov (plus transaction costs)	Comparison with B&H	FOREX – EUR/USD	2007-2010	878 pips (Avg. Profit incl. prov)
[41]	2020	GA, TA	Hybrid cost function – Sharpe Ratio and Sterling Ratio combined, Leverage	Sharpe Ratio, Sterling Ratio, Annualized Return	Comparison with B&H, S&H, GA-MR	FOREX – 6 major currency pairs	2018-2019	320% (Annualized Return for AUD/USD)

Table 2.1: Relevant works found in the literature.

Chapter 3

Proposed Architecture

This chapter presents the proposed architecture of a system whose purpose is to create an investment strategy to trade Foreign Exchange Market currency pairs in trending markets, characterized by using a strategy based on the channel pattern optimized by a Genetic Algorithm. The following Sections will provide a description of each module's implementation as well as its purpose.

3.1 System Architecture

A channel is a chart pattern that provides entry and exit points of a trade and it consists of two parallel lines (walls of the channel) that work as resistance and support levels: one linking swing highs and the other linking swing lows. It provides an interval for the expected price movement given its previous values and it is a useful and simple tool to trade in trending markets: when the price makes a series of higher highs and higher lows there is an uptrend so an ascending channel will form, otherwise if the price reaches consecutive lower highs and lower lows it will be a descending channel. When the market is sideways the channel is horizontal, indicating a zone of price consolidation. When the price breaks through the channel walls, it is said that the channel is broken and is no longer valid. The more volatile the market is, the greater the chances of catching big trends and consequently long channels however as the volatility increases the risk involved in the trade increases as well.

This thesis focuses on the development of a channel based trend following investment strategy, which means that it only trades when the market is trending, therefore approximately 60% of the time there are not any open positions, avoiding ranging market conditions which translates in less investment risk. The methods applied in the channel design are based on Price and Technical Analysis and the currency pair used to develop, train and test this model is the EUR/USD. Once the results of the algorithm are satisfactory, i.e. not entering a trade when the market is sideways and not entering a trade too late (when the trend is almost over), a Genetic Algorithm is applied aiming for profit maximization by optimizing the variables that define the channel.

Lastly, the system was implemented in Python due to its simplified syntax as well as the numerous and powerful libraries available to develop projects in the field of Data Science.

In order to fulfill the proposed system's goals, this work was divided into 3 modules (shown in Figure 3.1), each with a distinctive objective and responsible for an independent set of functions, executed in the order presented below:

- **Data Module** (Section 3.2): Module responsible for collecting the data from Dukascopy Bank and preparing it to be used by the Technical Rules Module. The preparation/pre-processing phase consists on cleaning and resampling the data.
- **Channel Module** (Section 3.3): Module responsible for creating the trading strategy. The strategy is based on the channels found and built, which are defined in this module.
- **Optimization Module** (Section 3.4): Module responsible for optimizing the trading strategy, using a Genetic Algorithm to improve the variables involved in the strategy definition.

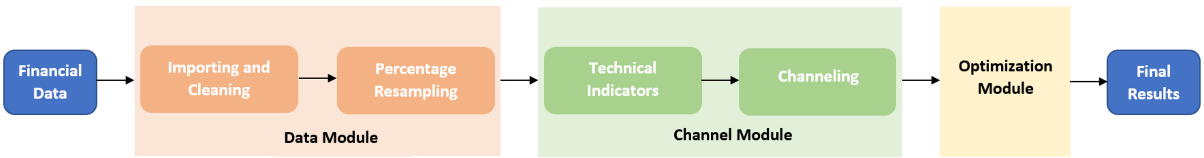


Figure 3.1: System Architecture.

Figure 3.2 provides a more detailed view of the architecture of the proposed system, after the data pre-processing phase. The channel module receives the data from the data module, after it has been cleaned, i.e. eliminating rows with empty columns or NaN values, and resampled by percentage, which means data samples were resampled according to their close price percentage variation (originally data was sampled by time intervals).

The first step in channel module is to apply the technical indicators zig zag and fibonacci pivot points. The dataset is traversed and for each sample these two technical indicators are calculated. Zig zag select the peaks and troughs in the dataset, which are then used as starting points to look for channels, since peaks and troughs provide price reversing points. Fibonacci pivot points are used to measure the risk involved in a trade. The moment a channel is found, a trading position is opened only if the price is at a pre-determined distance to a pivot point.

Then, in channeling sub-module is where the trading strategy is defined. *Finding Channels*, as the name implies, is responsible for identifying when the conditions to form a channel are met. Once these conditions are satisfied, *Building Channels*, based on the candles that formed the channel, will project channel walls and, until the price breaks these walls, the channel keeps being projected. When the price breaks through the source or target walls, the channel ends and the process of *Finding Channels* restarts. As the dataset is traversed, *Classifying the dataset* maps each sample with a Buy, Sell or Idle (i.e. if a position is opened, close it; if not, do not open a position) signal.

Throughout the development of this strategy, the parameters that tune each technical indicator, the resample percentage and the properties of a channel, are given arbitrary values. The genetic algorithm is used to fine tune these parameters, aiming to maximize the ROI.

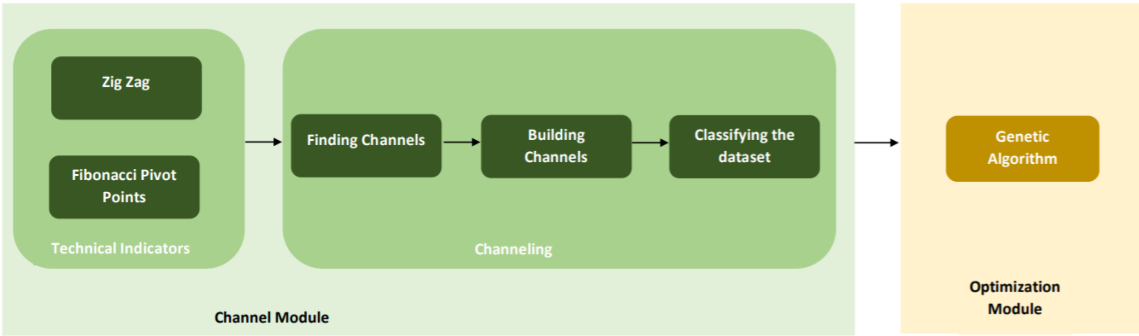


Figure 3.2: System Architecture in more detail.

3.2 Data Module

3.2.1 Importing and Cleaning

The first step on the development of this thesis proposed system is to get and prepare the data. First, the data is collected from Dukascopy: it consists on a structured time series dataset in OHLC (open-high-low-close) format of the EUR/USD Ask price and with a granularity of 1 sample per minute. The data is retrieved in a CSV file (comma-separated values), corresponding to the period from January 1, 2017 to December 31, 2017, which is stored in a dataframe containing 5 columns:

- Time: Open time for a time series data point with the format "YYYY-MM-DD hh:mm:ss". As mentioned earlier in this Section, each data point is separated by an interval of 1 minute.
- Open: Starting price of a currency pair for a given trading period.
- High: Highest price of a currency pair for a given trading period. Low: Lowest price of a currency pair for a given trading period.
- Close: Closing price of a currency pair for a given trading period.

In a first phase, the method to collect the data from Dukascopy was by coding: we would download the tick data, which is the most granularly high-frequency data available, and then resample it with a granularity of 1 minute intervals in a OHLC format. Unfortunately, this method proved to be inefficient because downloading files corresponding to 1 year of tick data would produce massive files and most of the times with a lot of missing data, just to be resampled later to a 1 minute timeframe. Even though this process could be changed to download data already spaced by the desired interval, a tool called Tickstory Lite was used to perform this step. It has proven to be faster, with a very friendly interface (presented in Figure 3.3), several pre-defined output formats (for example CSV files with a generic bar format) and customizable output format. Furthermore, it returns consistent and complete data.
The next step is to clean the dataset. Since the data is accurate, the only concern is to guarantee its completeness, i.e. that no missing data is found in any rows of the dataset.

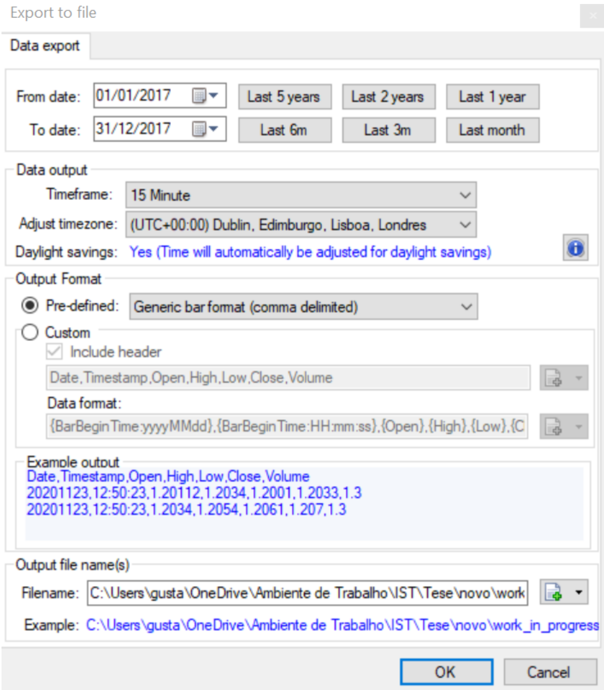


Figure 3.3: Interface of Tickstory Lite for exporting currency pair historical data.

3.2.2 Percentage Resampling

The standard and most popular way to organize a dataset is by dividing the samples according to a fixed time interval, i.e. by time sampling. This is not ideal, since markets do not process information at a constant time interval. Time bars oversample information during low activity periods and undersample information during high activity periods [42].

That said, in this work the dataset is resampled according to the closing price percentage variation, thus obtaining a more homogeneous dataset. Simply put, the samples of the original dataset grouped in blocks, being those blocks the samples of the resampled dataset. In more detail, as algorithm 1 shows, the dataset is traversed and consecutive data points are continuously aggregated forming a block. The Open (price) of the first sample aggregated defines the Open (price) of the block and the Close is given by the Close (price) of the last sample added. The High and Low of the block correspond, respectively, to the highest and lowest values of the samples aggregated. Whenever the percentual difference between the last and the current blocks Close (prices) exceeds the threshold defined to 0.02%, the current block is “closed” (forming a data point that is going to be appended to the resampled dataset) and a new empty block is generated. It is possible that a block is composed by only 1 data point of the original dataset, in the case of high market activity registered in that period. Despite the resampling method chosen was based on percentage price variation, other processes could have been applied, resorting to different properties, such as the amount of price variation (measured in pips) or the volume of the currency pair trades.

Figure 3.4 presents the dataset before and after the resampling process.

Algorithm 1: Resampling Algorithm, based on close price variation.

```
resampledData = dataframe to store resampled data;
currentBlock(Open,High,Low,Close) = first dataPoint of dataset;
testClose = closing price of the first dataPoint of dataset;
resetFlag = 0;
for each dataPoint in dataset do
    if resetFlag then
        resetFlag = 0;
        reset currentBlock, which means currentBlock(Open,High,Low,Close) = dataPoint;
    end
    if dataPoint(Low) is lower than currentBlock(Low) then
        update currentBlock(Low) with dataPoint(Low);
    end
    if datapoint(High) is higher than currentBlock(High) then
        update currentBlock(High) with dataPoint(High);
    end
    if the variation between dataPoint(Close) and testClose is superior to 0.02%: then
        resetFlag = 1;
        update currentBlock(Close) with dataPoint(Close);
        append currentBlock to resampledData;
        update testClose with currentBlock(Close), currentBlock(Close) becomes the term of
          comparison for price variation;
    end
end
```

(a) Before resample

(b) After resample

Figure 3.4: Candlestick chart of the dataset.

3.3 Channel Module

The Channel Module is responsible for implementing the investment strategy, through which the classification and forecasting of the market behaviour is made. In order to do so, it receives the pre-processed data from the Data Module. This module is composed by 2 components:

- **Technical Indicators**: Extract relevant features from price, playing a preponderant role in the construction and decision-making of the investment strategy.
- **Channeling**: Establishes the investment strategy itself. Builds the channels throughout the dataset and classifies each sample as Buy, Sell or Out.

This may be considered the most relevant module of this thesis, since it's where the trading strategy is defined. The previous module (Data Module) goal was to prepare the data to be consumed by the Channel Module and the next one's (Optimization Module) purpose is to refine the strategy developed in the Channel Module.

3.3.1 Technical Indicators

Two technical indicators were found relevant for the definition of the trading rules in the investment strategy of the proposed solution: Zig Zag Indicator and Fibonacci Pivot Points.

A detailed description of these indicators is given in Section 2.1.3, but this Section serves the purpose of giving a contextualization of their utility in the investment strategy algorithm.

Zig Zag

The Zig Zag is a lagging indicator that plots points on the chart whenever prices reverse by a percentage or a magnitude greater than a pre-chosen variable. Straight lines are then drawn, connecting these points. By filtering out insignificant price fluctuations, it identifies swing highs and swing lows. In this solution, the function that reproduces the zig zag indicator receives the dataset from the Data Module as input, identifies all the points where there is a price reversal, select from these points the ones whose price differ more than 3 pips respecting the condition that after a swing high must come a swing low and finally returns a list of tuples, where each tuple is composed by 2 values: the first one is the corresponding index of the zig zag point in the dataset and the second one identifies if it is a swing high or swing low, by storing the value “peak” in case of a swing high and the value “valley” in the case of a swing low. Each zig zag point will be the starting point for the search of a channel. The reason to do so, is that these points identify the beginning of trends and the goal in a trend following trading system is to identify the trend as soon as possible and then “ride it”. As referred above, during this development stage, the value chosen as a filter for the indicator was 3 pips. Even though it is expressed in magnitude, this parameter could have been implemented as a percentual difference, as was done in Section 3.2.2 to define the resampling threshold, and its value would be approximately 0.02%. The zig zag indicator functioning is illustrated in Figure 3.5.

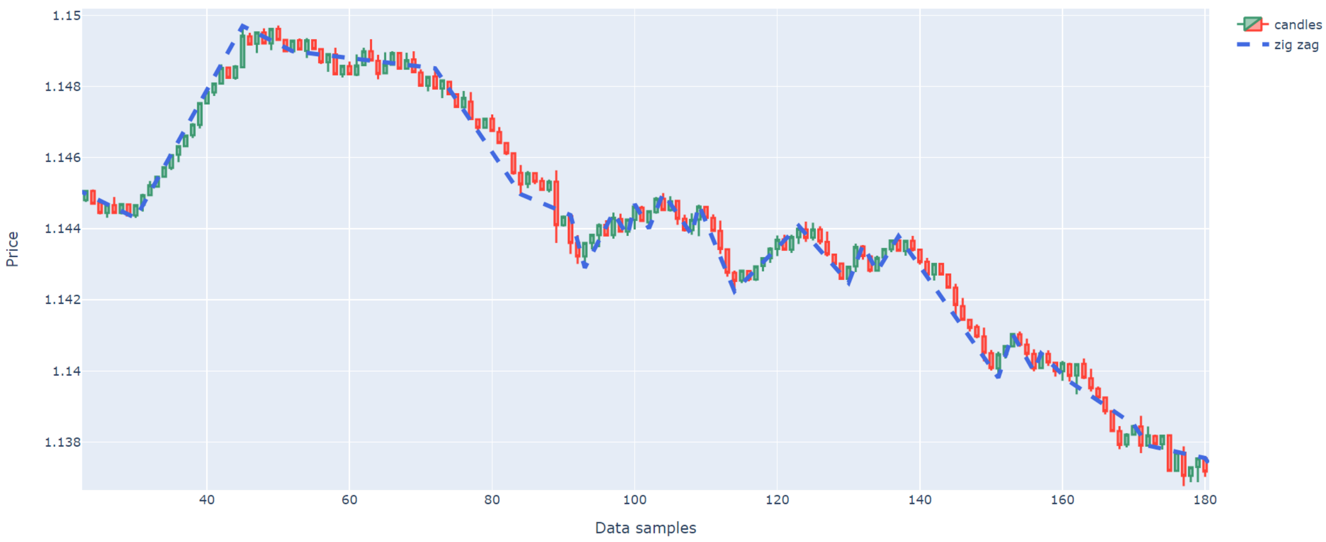


Figure 3.5: Application of the Zig Zag technical indicator.

Fibonacci Pivot Points

Pivot Points is a leading indicator that, based on the High, Low and Close prices of the previous trading day, calculates possible support and resistance levels. Many traders like to incorporate Fibonacci Ratios in their strategy, especially to calculate Fibonacci Retracements and Extensions, which are other methods to predict support and resistance levels when the market is trending. That being said, since this solution proposes a trend following strategy and that we want to go in the same direction as the market sentiment, Fibonacci Ratios are taken into consideration for the calculation of the Pivot Points.

Upon their calculation, a new column containing its values is added to the dataframe that stores the dataset. Their purpose is to prevent the trader from opening a position when the trend is almost over. More precisely, if a position is about to be opened when the price has travelled more than 90% of the distance between two pivot points (these two correspond to the nearest pivot points that are above and below the price), the position is not opened because the price is too close to a pivot point where a trend/price reversal may occur. Figure 3.6 shows an example a possible channel formation during an uptrend: the arrow marks the candle where a channel would begin; since this candle is located in the green area, a position could be open, otherwise (if it was located in the red area) the candle would be too close to the next pivot point (yellow line), hence none position would be open.

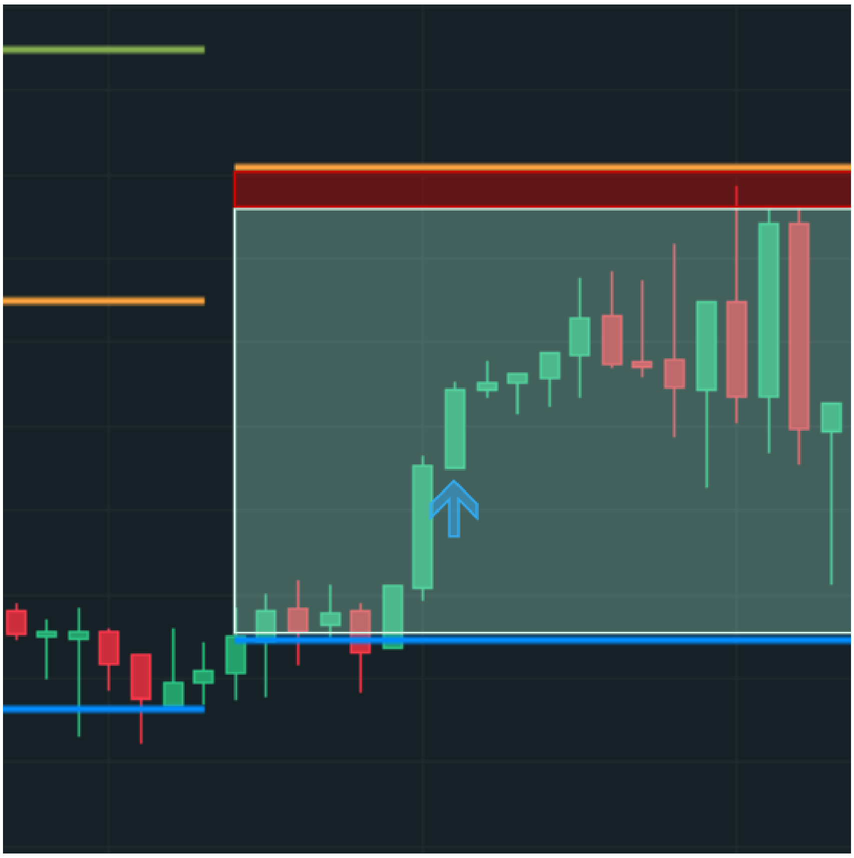


Figure 3.6: Example of the Fibonacci Pivot Point role in channeling.

3.3.2 Channeling

This second component is responsible for implementing the trading strategy and, in a broad overview, it can be divided in 3 functionalities:

1. Finding a channel.
2. Building a channel.
3. Classifying the dataset.

These 3 functionalities are not independent blocks where the generated output of each of them enters the next block as the input, for example at the same time that the dataset is being iterated searching for channels, the classification of each sample iterated is being made. However, it is possible and provides more clarity of each function, if they are presented as 3 separate blocks.

Finding Channels

Finding a channel is the process of identifying the place where a channel can be built/drawn (in a trading platform channels can drawn by hand), an overview about the channel pattern is provided in Section 3.1.The flowchart in Figure 3.7 should be seen as guide for the explanation given forward, since it gives the big picture of the algorithm developed to find channels. Table 3.1 provides a description of the variables employed in Figure 3.7

Parameters	Description
z	index of current zigzag point
d	index of current dataset sample
ZZ	current zigzag point
c	counter, counts 4 consecutive points where closing price is consecutive higher/lower
CP	closing price - current/previous if it’s the closing price of the current/previous data sample
Too close to enter	If closing price is too close to Fibonacci Pivot Point

Table 3.1: Variable description of finding channel flowchart.

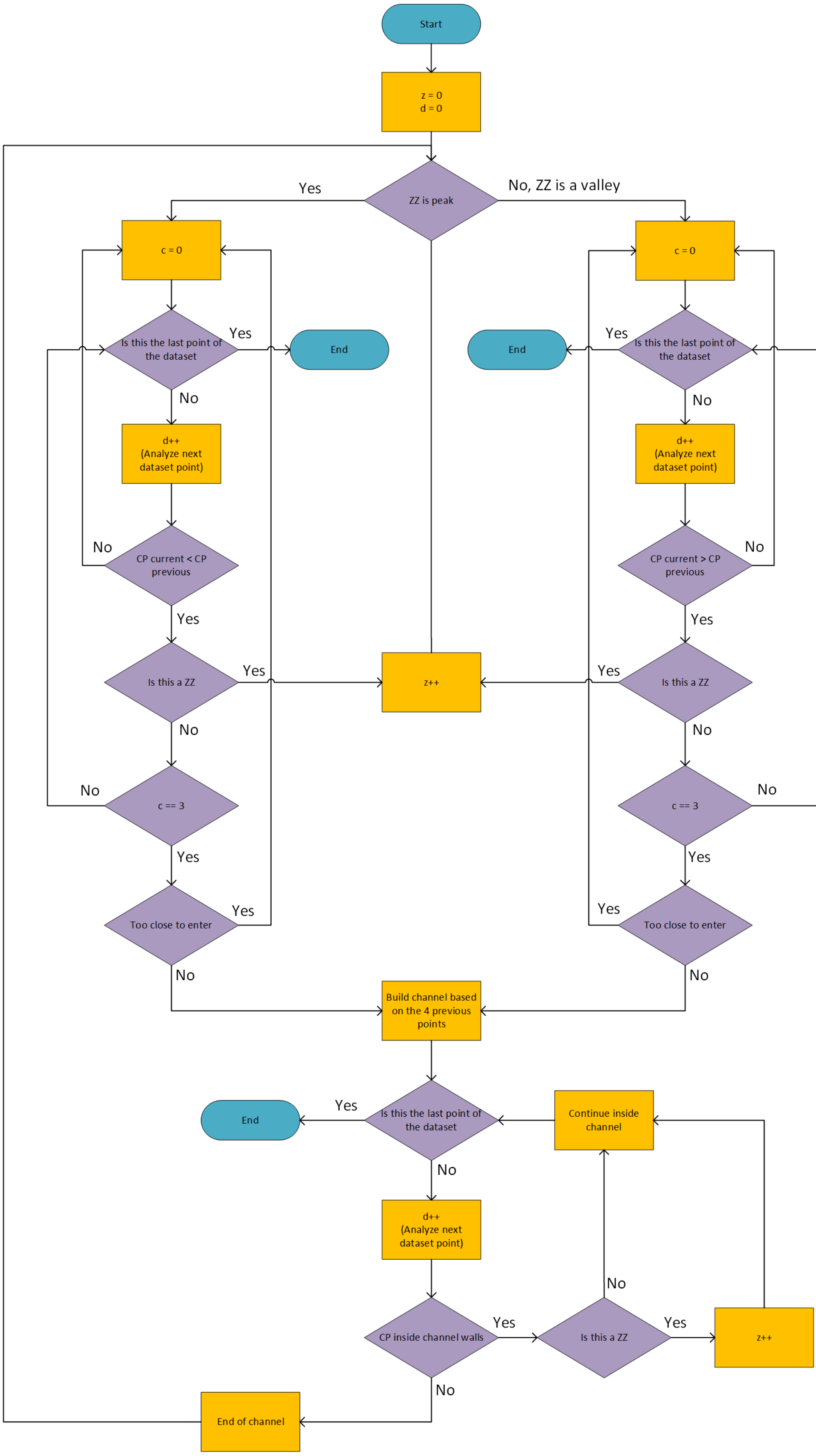


Figure 3.7: Finding channels flowchart.

As stated before, the aim is to only trade in trending markets, hence a set of rules that ensure that the algorithm does not miss trends and, at the same time, avoids sideways conditions must be defined. By try and error (i.e. after applying one strategy, inspect the resultant graphic with price all channels drawn and check if it fits our demands), 2 strategies performed equally well:

- Simple Moving Average: Find 4 consecutive samples in the dataset where the sample SMA value is higher than the previous sample SMA value (ascending channels), if we are in an uptrend. In the case of a downtrend the SMA of the sample must be lower than the previous (descending channels).
- Closing Price: Find 4 consecutive samples in the dataset where the sample closing price is higher than the previous sample closing price (ascending channels), if we are in an uptrend. In case of a downtrend the closing price of the sample must be lower than the previous (descending channels).

The choice fell on the second option, comparing by closing price.

Finding a channel means that we have a trend confirmation because, with trend following strategies (like this one), we're only able to spot the trend after it has began. So, we need a way to identify a trend as soon has possible. This component receives as input the dataset and the list with the zigzag points (list containing the index of swing highs and swing lows), both coming from the component Technical Indicators, which is described in Section 3.3.1. As Figure 3.8 illustrates, the zigzag list contains the position (index) of the swing highs and swing lows in the dataset and the information that tells if that specific sample is a swing high or a swing low. Therefore, these samples correspond to price reversals, in other words they mark the starting point of possible trends. Hereupon, the dataset is iterated, starting on the index value of the first element of the zigzag list, and guiding the search of the type of channel by the index value of the current zigzag point, being the following procedure adopted:

1. Analyze the current zigzag point, getting its values: index in the dataset, which will be called just index until the end of this procedure description; and the information regarding if it a swing high or a swing low.
2. Search for channels in the dataset, starting on the index kept by the current zigzag point until the index of the next zigzag point in the list is reached (the next zigzag point corresponds to a trend reversal because the market moves in waves, so after a swing high comes a swing low). If the current zigzag point is a swing low, we are in the presence of an uptrend, hence we look for ascending channels. On the other hand, if it is a swing high, we look for descending channels.
3. If no channels were found (exemplified in Figure 3.9), update the current zigzag point with the next zigzag point and restart this procedure. If a channel was found, continue iterating the dataset until the price breaks through the channel walls. Whenever the data sample being analyzed corresponds to a new zigzag point, update the current zigzag point.
4. When the channel finishes, restart this procedure starting on the data sample where the channel was broken: if the next zigzag point has not been reached yet, the zigzag point has not been

updated, therefore the algorithm continues looking for channels in the same direction (exemplified in Figure 3.10), starting from the data sample that broke the channel. On the other hand, if the next zigzag point(s) was/were overtaken, update the current zigzag point and according to its nature look for ascending or descending channels, starting from the data sample that broke the channel(exemplified in Figure 3.11, the arrows point to the 2 next zigzag points included in the first channel).

If the current zigzag point being analysed is the last point of the list, the algorithm will be run until the end of the dataset.

Furthermore, in the interest of minimizing the risk of every trade, Fibonacci Pivot Points are used. As explained in Section 3.3.1, these provide possible areas of price/trend reversal. Therefore, in the proximity of a pivot point the chances of getting fake signals (i.e. the conditions to form a channel are met but quickly the price reverses), are higher. In this solution, after a channel is found, if the price has travelled more than 90% of the distance between the two nearest pivot points that are above and below the price, we consider that the price is too close to the upcoming pivot point and that channel formation is ignored. Figure 3.12 shows an example, where the fibonacci pivot points are the blue and yellow lines surrounding the region being analysed. After a swing low (pointed by the red arrow) we have 4 consecutive samples with a closing price higher than the previous. Since the closing price of the last candle (pointed by the blue arrow) is inside the red area, which means the price is too close to the upcoming fibonacci pivot point, the channel is not created.

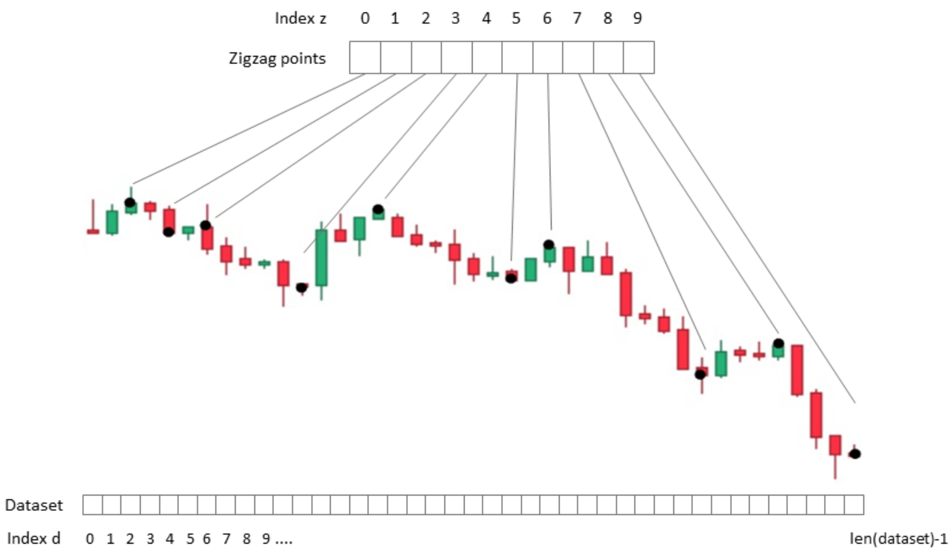


Figure 3.8: Representation of the relation between the zigzag list and the dataset.

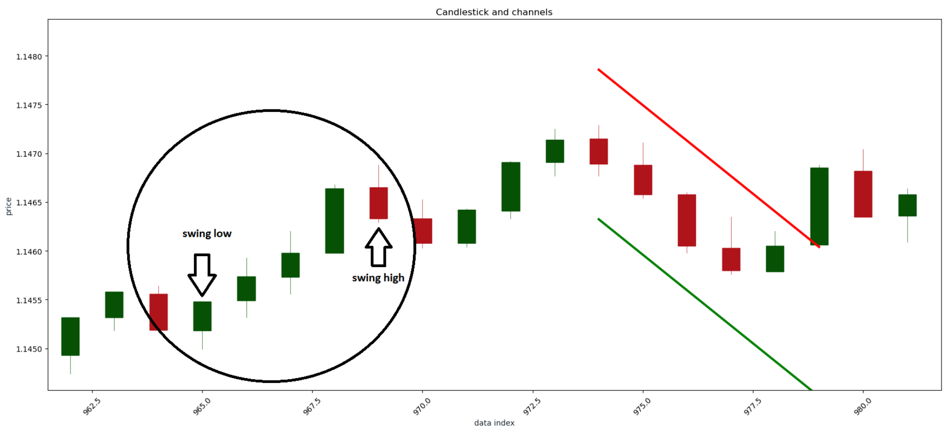


Figure 3.9: Example of iterating between 2 zigzag points without finding a channel.

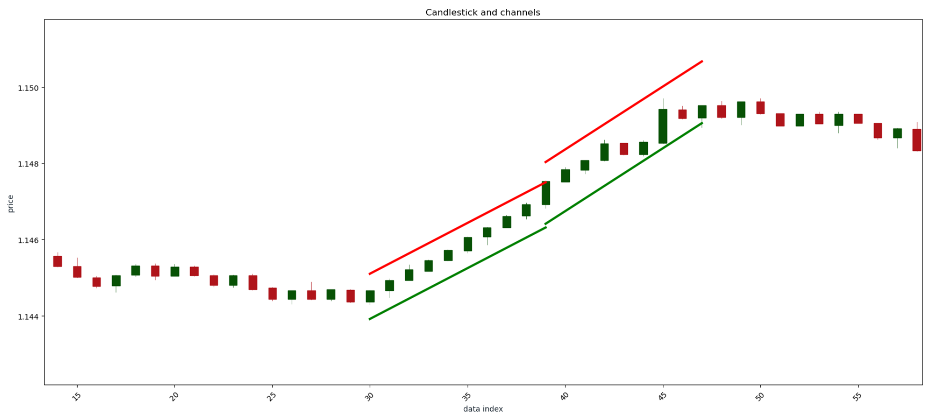


Figure 3.10: Example of finding 2 channels: the second is formed before the next zigzag point is reached.

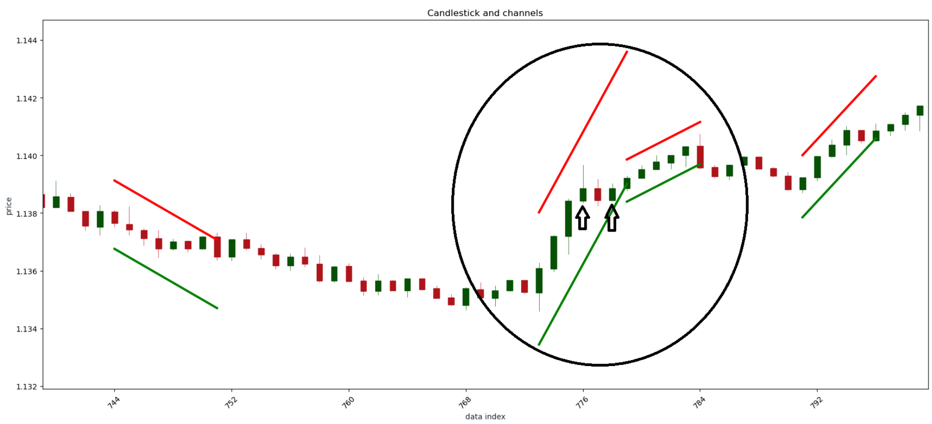


Figure 3.11: Example of 2 channels, where the first one included 2 zigzag points and the second one starts in the data sample that broke the channel.

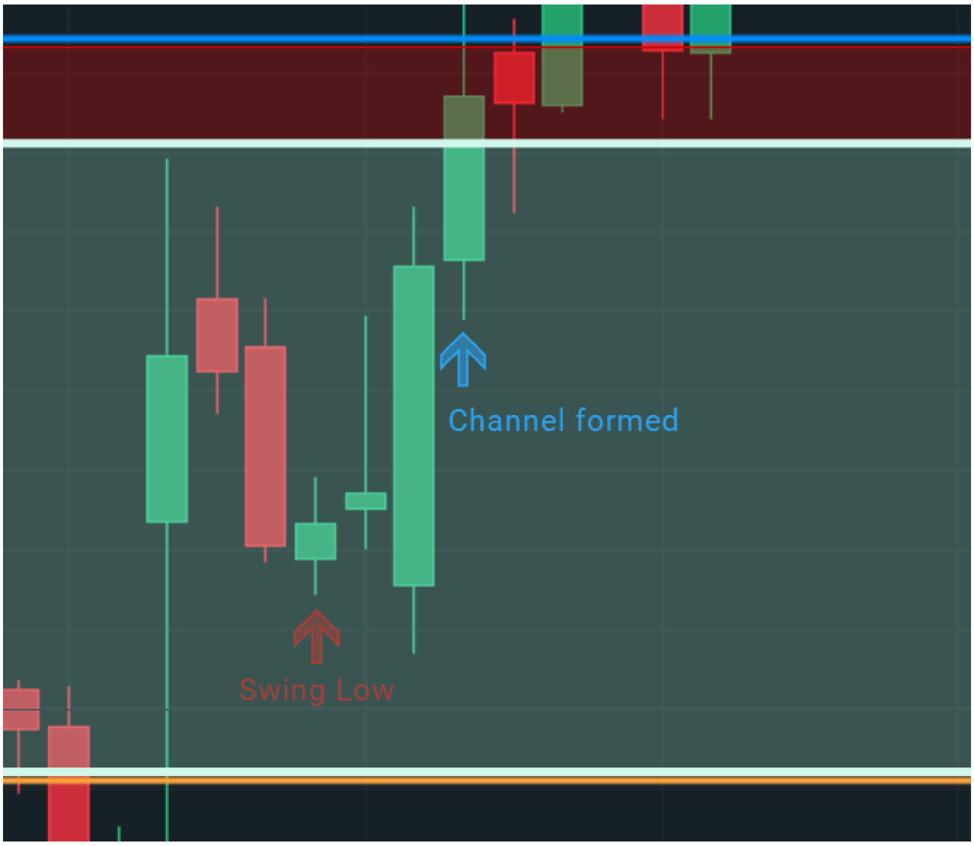


Figure 3.12: Example of a channel found too close to a Fibonacci Pivot Point.

Building Channels

Building a channel is the process that takes place once a channel is identified and, as mentioned in Section 3.1, consists on defining two parallel lines that constitute its walls. The target wall is the parallel line that defines the upper limit of the channel and the source wall is the parallel line that defines the lower limit of the channel. The channel remains valid while the price moves within its walls. Once the price breaks them, the procedure to find a new channel restarts, as explained in Section 3.3.2.

Building a channel is implemented as follows:

- First, calculate the slope of the channel walls. As said before, 4 samples are necessary in order to find a channel. Having said that, the slope of the channel walls is given by the slope of the line that passes by the closing price of the first and the last (fourth) samples, as the reader may confirm in Figure 3.13, where the slope between the first and the fourth sample is depicted by a black line.
- Then, the intercept parameter of the line equation is adjusted in order to encompass the 4 candles. That is, ensure that the channel includes the high and low prices of these 4 candles that are used to create the channel. This would yield a channel like the one shown in Figure 3.14.
- Finally, one final step is taken to improve the channel. At this point, the way a channel is built makes it vulnerable to small random price variations because the channels are very narrow. So, even if the trend is very strong, any variation of the price in the opposite direction of the trend will break the channel walls. To overcome this problem, additional width is provided to the channel.

A parameter, which was named *plus channel width*, regulates how much more width is given and initially (during the development phase) a value of 0.5 was established, but this parameter is fine tuned in the Optimization Module. Increase the channel width by 0.5 means that both walls are pushed away $0.5 \times original\ channel\ width$ or 50%, as may be observed in Figure 3.15.

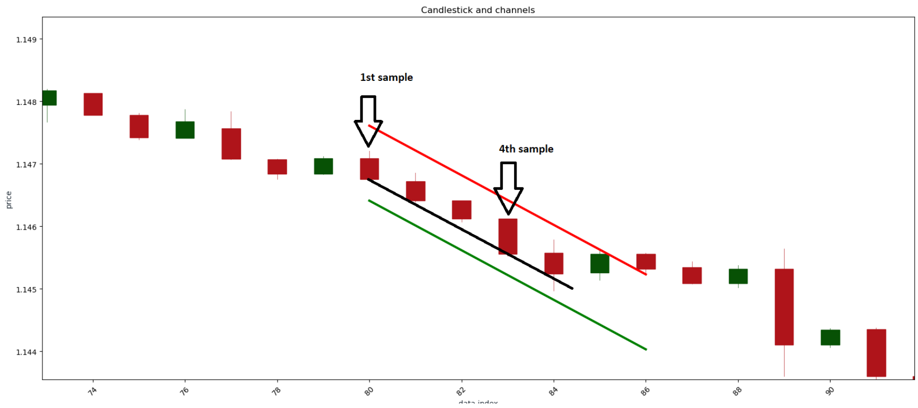


Figure 3.13: Demonstration of the method used to calculate channel slope.

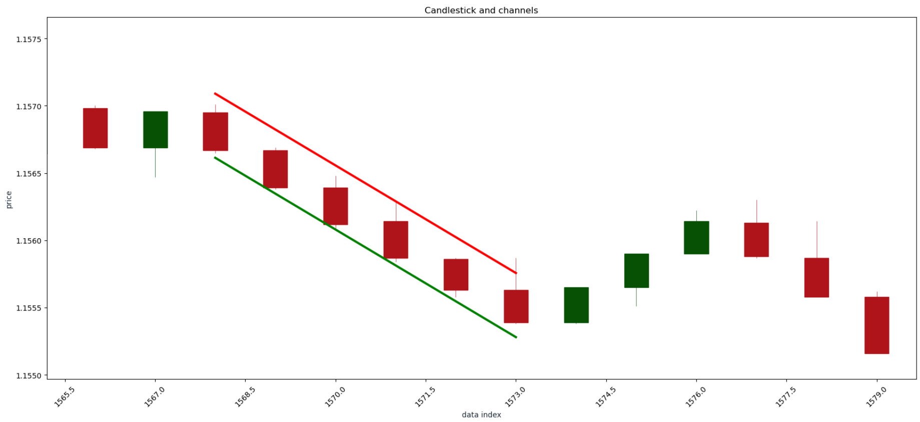


Figure 3.14: Example of the interception parameter adjustment.

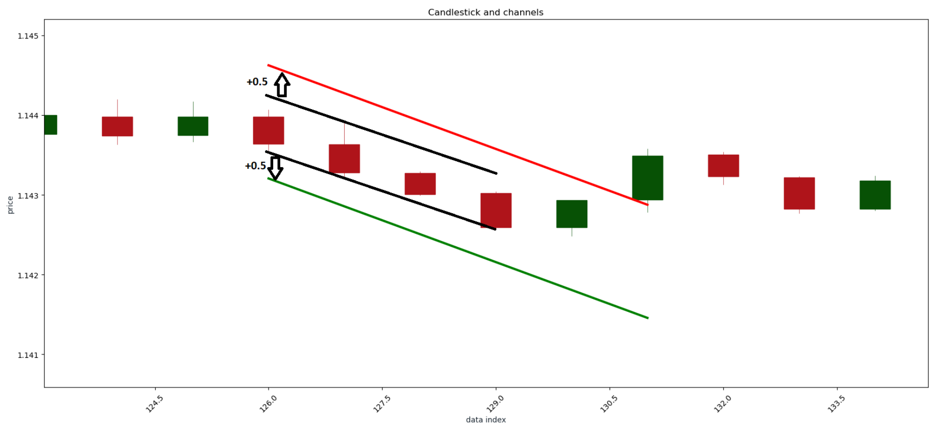


Figure 3.15: Example of the channel width adjustment.

Classifying the dataset

Classification is the final step to complete the trading strategy. As the dataset is being iterated searching for channels, each sample is being classified, having 3 possible values:

- **0**: If a sample is not inside a channel it's classified with the value 0 and produces the signal Out, which means do not open a position or, if a position is open, close it.
- **-1**: If a sample is inside a descending channel it's classified with the value -1 and produces the signal Sell, which means open or maintain a selling position.
- **1**: If a sample is inside an ascending channel it's classified with the value 1 and produces the signal Buy, which means open or maintain a buying position.

3.4 Optimization Module

This module is responsible for the optimization of the parameters that characterize this investment strategy by using a Genetic Algorithm, which has been introduced in 2.1.4, a search algorithm based on the Darwin's theory of natural selection.

It starts with a random initial population (set of chromosomes), where each individual (chromosome) represents a possible solution to the problem. Then, based on the evaluation of the quality of each solution (Fitness Function), these individuals mate (Selection), reproduce (Crossover and Mutation) and evolve until some termination criteria is met. Finally, the best individual found (the one whose Fitness Function value is the highest) corresponds to the optimized solution.

3.4.1 Chromosome Representation

As explained in 2.1.4, each individual in a population is a chromosome that, in turn, is composed by a set of genes. In this solution, each chromosome is represented by a list of 5 genes:

- **Number of candles**: Number of candles necessary for a channel to be formed. It is an integer value and can vary between 1 and 10. In Section 3.3.2 this value had been predefined to 4 candles.
- **Additional channel width**: As the name tells, it is the additional width given to a channel. It is represented by a float with 2 decimal digits and can vary between 0.00 and 1.00 or 0% and 100%. In Section 3.3.2 this value had been predefined to 0.5 or 50%.
- **Percentage resampling**: Percentage threshold defined when rearranging the original financial data. It is represented by a float with 2 decimal digits and can vary between 0.00 and 0.50 or 0% and 50%. The reason why this gene upper boundary is ”only” 50% is that 50% of percentage variation corresponds to a variation of approximately 70 pips in magnitude between each simple, which already is a very high value. In Section 3.2.2 this value has been predefined to 0.02%.
- **Zig zag filter**: Filter of the zig zag technical indicator used in Section 3.3.1, where it has been predefined to 3 pips. It is represented by an integer and can vary between 0 and 50 (pips).
- **Pivot Point filter**: Filter of the Fibonacci Pivot Point. As explained in Section 3.3.1, a channel is ignored when formed above the level of this filter. It is represented by a float with 2 decimal digits and can vary between 0.00 and 1.00 or 0% and 100%. It has been predefined to 90%.

Table 3.2 summarizes the chromosome representation.

Gene	Number of candles	Additional channel width	Percentage resampling	Zig zag filter	Pivot Point filter
Genotype representation	int	float	float	int	float
Interval	[1,10]	[0.00,1.00]	[0.00,0.50]	[0,50]	[0.00,1.00]

Table 3.2: Chromosome representation.

3.4.2 Fitness Function

The fitness function measures, quantitatively, how good each individual of the population is and the probability that an individual will be selected for reproduction is based on its fitness score. The fitness function used in this solution is the *Return on Investment* (ROI), where the goal is to maximize it. Therefore, the greater the ROI value the fitter the individual. Briefly, it is calculated in the following way:

$$ROI = \frac{Returns - Investment}{Investment} \tag{3.1}$$

For every trade, a fixed spread value of 1 pip was used as well as a fixed position size of 1 standard lot (that is equivalent to 100000 units).

3.4.3 Population Size

The more complex the problem the bigger the population should be. Most of the GA solutions found tune this parameter to 100 or more. However, since each chromosome only has 5 genes, the population size was set to 50, which correspond to 10 times the number of genes, already providing a substantial degree of diversity to the population.

3.4.4 Operators

Selection

Ideally, we want our population's overall fitness to increase, and selection helps us to do so by choosing the best-suited individuals and discarding those who are not well adapted to solve the target problem. Without it, a GA algorithm is only a set of methods that generates different solution each time it runs. There are various ways to perform the selection, but the main idea is to increase the chances for fitter individuals to be preserved over the next generation. However, if the selection pressure is too high (selection pressure is the degree to which the better individuals are favored) the diversity of the population is lost and the solution may converge to a local optimum.

That being said, 2 types of selection were selected: Elitist Selection and Tournament Selection. Elitist Selection is based on the principle that the best individuals of the current generation survive to the next generation, preventing the lost of the best solutions. It must be used with care, since it may lead to lost of diversity leading to a premature convergence of the solution. In this solution, the 10% most fit individuals survive to the next generation by Elitist Selection. The disadvantage is that it requires the population to be sorted first, according to its individuals fitness values.
The other selection method chosen applies to the remaining 90% of the population, it is the Tournament Selection. Both parents are selected using Tournament Selection, where k individuals are randomly selected from the population. Then, the best out of these is chosen to be a parent for the next generation. The selection pressure is adjusted by varying the variable k, being that if k is equal to 1 the parent selection is totally random and as it is incremented the selection pressure increases. In this solution, k=3. Furthermore, Tournament Selection allows the usage of negative fitness values, which may occur in the proposed system.

Crossover

Crossover is the method that mimics the natural reproduction. The idea is that by combining 2 already fit individuals, the offspring will be even fitter, by inheriting genes from each parent.

In this work, the crossover method adopted is Uniform Crossover because, as discussed in 2.1.4, in the long term Uniform Crossover becomes less biased and the search is more global than, for example, Single-point Crossover, where in the beginning a lot of genetic material is swapped between the parents, but rapidly convergences to local solutions.

In Uniform Crossover each gene of a parent chromosome has a 50% chance of being swapped with its correspondent of the other parent, giving origin to 2 child chromosomes, as illustrated in the example of Figure 3.16. For each gene of the chromosome a random integer number between 1 and 100 is generated and: if this number is inferior or equal to 50 that gene is swapped with the correspondent gene of the other parent, otherwise none exchange occurs.

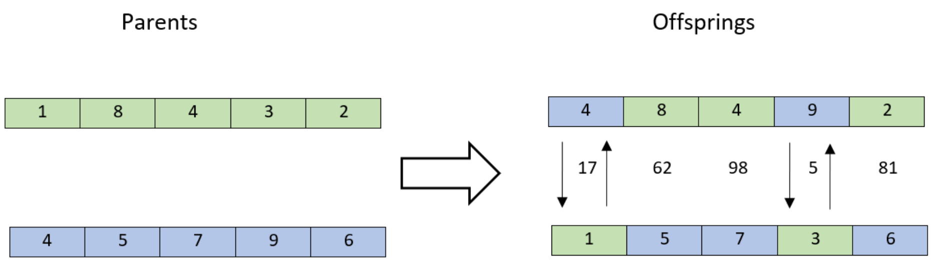


Figure 3.16: Example of Uniform Crossover operation.

Mutation

The offspring generated during crossover are then subject to mutation. Mutation is a process used to maintain genetic diversity in the population. Without it, all the combinations that we would ever possibly reach during the successive generations would be already in the initial population.

Among the several mutation operators commonly used, the one applied in this work is called Random Resetting. It is applied as follows: given a chromosome, each gene has a certain probability of being replaced by a random value from the set of possible values for that gene. This probability is called mutation rate and was defined to 10%. Summing up, for each gene of the chromosome a random integer number between 1 and 100 is generated. If this number is inferior or equal to 10, that gene is swapped for a random value belonging to the set of possible values for that specific gene.

3.4.5 Random Immigrants

After undergoing Selection, Crossover and Mutation there is only one final step to create a new generation: Random Immigrants [34]. Random Immigrants is another method responsible for introducing diversity in the population by replacing the least adapted individuals of the population for new and randomly generated individuals. We will call the variable that define the portion of individuals replaced Immigrant Rate, which was defined to 20% and is fixed value. Therefore, for every generation and after mating and reproducing, the individuals are sorted by their fitness value and the 20% least fit individuals are replaced by random ones. Now, the new generation is complete.

3.4.6 Termination Criteria

The GA stops if one of two conditions is met:

- The algorithm completes 100 generations. After 100 generations it is assumed that the algorithm has already converged to best solution found.
- The population does not evolve, i.e. the fitness value of the best individual found so far does not increase, during 10 consecutive generations. If the algorithm converges before 100 generations, there is no need to run the algorithm until it completes 100 generations.

3.4.7 GA Parameters and Flowchart

The flowchart of the GA is presented in Figure 3.17, as well as the parameters used in the GA which can be seen in table 3.3. The algorithm has the disadvantage of needing to be sorted twice. The first occasion is when Random Immigrants are introduced. The second is before the Selection operation, since the Immigrants are appended to the end of list of chromosomes without taking into consideration its fitness value, therefore the population needs to be sorted to get the best 10% individuals.

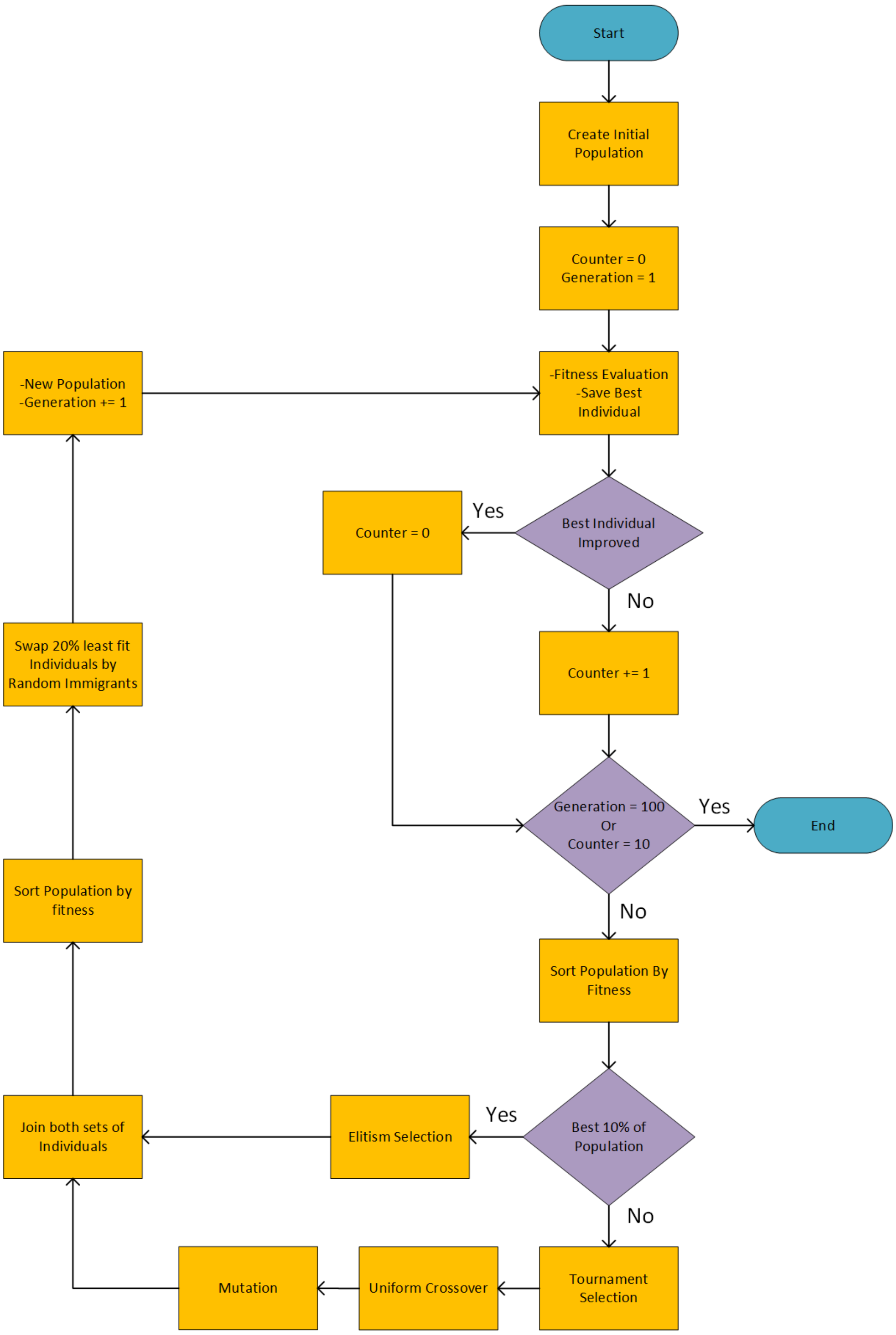


Figure 3.17: Genetic Algorithm Flowchart.

Parameters	Value
Chromosome Size	5 genes
Population Size	50 individuals
Elitism Rate	10%
Tournament Selection - Selection Pressure	k = 3
Mutation Rate	10%
Random Immigrants Rate	20%
Termination Criteria	100 generations or 10 generations without evolution

Table 3.3: Genetic Algorithm parameters setting.

3.5 Chapter Conclusions

In this chapter a full description of the developed solution is made. It provides an insight of the implementation decisions made, equipping the reader with the necessary tools to develop his/her own solution or extend the one presented. This work architecture is divided in 3 independent main modules: Data Module, Channel Module and Optimization Module. These, in turn, are subdivided in several tasks, which, put together, compose the trading strategy. All of the parameters that define the behaviour of the channels, which are optimized in the Optimization Module with a GA, are customizable.

In chapter 4, the system will be evaluated.

Chapter 4

System Validation

This chapter is intended to evaluate the quality of the solution proposed in this thesis (discussed in 3). First, the metrics used to measure its performance are introduced. Then, in order to verify the quality of the proposed system and the tuning of the parameters used in this system, 4 case studies are discussed:

1. Studies the approach taken regarding the trading strategy, more specifically the use of Fibonacci Pivot Points.

2. Studies the influence that GA parameters have in the system's performance.

3. Compares the system's performance when trading different currency pairs: one major pair, one minor pair and one exotic pair.

4. Presents the best performing results.

Since the first three case studies focus on comparing the quality of the solution by using different tuning values rather than achieving the highest return, only one year of data was used to train the model. The last fourth case study presents a solution in which the model is trained with two years of data.

It is important to note that the charts are separately displayed for each solution, instead of showing in the same chart the ROI for all solutions, for instance. Since the data is not sampled by a fixed time interval (because it was previous resampled according to the close price variation, as is explained in 3.2.2), different solutions have different number of samples making it difficult to interpret the charts. The only exception is in Case Study D, where the same chart contemplates the ROI of all tested years.

4.1 Evaluation Metrics

In order to evaluate the performance of the proposed solution and to be able to make comparisons between different approaches, the following measures were taken into account.

4.1.1 Return on Investment

Return on Investment (ROI) is a performance measure used to evaluate the profitability of an investment, by measuring the amount of return on an investment relative to its cost. ROI is expressed in percentage and it is calculated by dividing an investment's net profit (or loss) by its initial cost[43]:

$$ROI = \frac{Current\ Value\ of\ Investment - Initial\ Value\ of\ Investment}{Initial\ Value\ of\ Investment} \tag{4.1}$$

If an investment's ROI is positive, then it is worth investing. When comparing two investments, the one with a higher ROI is going to be the most profitable.

4.1.2 Sharpe Ratio

Sharpe Ratio is a metric that evaluates the risk-adjusted profitability of an investment. Its calculation is given by:

$$Sharpe\ Ratio = \frac{ROI - RF}{\sigma} \tag{4.2}$$

Where:

- ROI corresponds to the return of an investment, as described in 4.1.1.
- RF is the risk-free rate, which is the theoretical return of an investment with zero risk associated. The risk-free rate used in this work is 3.5% (of ROI).
- Standard deviation, represented as σ, introduces the component of volatility measure in the formula.

The higher the Sharpe ratio is, the better the investment/strategy is considered to be. Conversely, the lower the Sharpe Ratio is, the riskier the strategy is likely to be, and consequently probably also the less profitable over the long-term.

4.1.3 Other Metrics

- **Maximum Drawdown (MDD):** Measures the maximum lost in terms of accumulated ROI%, by capturing the greatest movement from a peak to a trough.
- **Percentage of profitable trades:** Given the total number of opened positions, how many were profitable.
- **Average profit per trade:** Average profit(loss), provided in *ROI%*. The higher the value the better.
- **Percentage of time in the market**: Measures how active in the market a trader (an intelligent system in this case) is.
- **Transaction costs:** Sum of transaction costs, in pips, throughout the test period.

4.2 Case Studies

This Section presents several case studies, including the comparison of several configurations of the trading strategy and the GA, as well as trading with different currency pairs.

4.2.1 Case Study A - Comparing different strategy approaches

As explained in 3, the proposed trading strategy has two components: the main one being the channels and the other being the Fibonacci Pivot Points (reviewed in 3.3.1), which are there to manage risk by avoiding opening positions too late, i.e. when the price has a considerable chance to change direction. That being said, in this case study the aim is to understand the influence of Fibonacci Pivot Points by comparing the strategy results with normal Pivot Points and without Pivot Points assistance at all. Normal Pivot Points distinguish themselves from Fibonacci Pivot Points by not incorporating Fibonacci ratios in their calculation.

The strategy simulation was conducted for the EUR/USD currency pair, with a data set of one year of data whose data frequency is one sample per minute. Independently of the market conditions, it is assumed a fixed spread of one pip and a fixed position size of one standard lot, as table 4.1 shows.

Simulation Parameters	Value
Market	Forex – EUR/USD
Training Period	2017
Test Period	2018
Position Size / Lot	100000 units / 1 Standard Lot
Spread	Fixed to 1 pip

Table 4.1: Case Study A - Simulation Parameters.

The GA parameters remained unchanged throughout the simulation. The values applied can be seen in table 4.2 and they remain the same as they were defined in Section 3.4.

GA Parameters	Value
Population Size	50
Elitism Rate	10%
Tournament Selection – Selection Pressure	k = 3
Mutation Rate	10%
Random Immigrants Rate	20%
Crossover Rate	50%
Termination Criteria	100 generations or 10 generations without evolution

Table 4.2: Case Study A - GA Parameters.

Having introduced the initial configurations, a brief analysis is made concerning the training of the system. During the training phase, the *GA* aims to find the best fitted individual among all individual throughout all generations. The features that are optimized by the algorithm are summarized in 3.4.1. Table 4.3 presents the training feature setup for the best individual found, for the 3 strategies compared. For simplicity, each strategy will be mentioned by its initials - FPP (Fibonacci Pivot Points), NPP (Normal Pivot Point), NonePP (No Pivot Points):

- **Nº generations to converge:** The 3 strategies converged due to being 10 generations without evolve, i.e. in 10 generations they didn't find an individual whose configurations enabled a ROI better than their previous best individual. NPP was the first to converge and, since the algorithm ended after being 10 generations without evolve, this means the best solution was found on the tenth (10th) generation. NonePP was the last to converge.
- **Nº candles:** The 3 strategies considered that, in order to admit the presence of a channel, they only need 2 candles (more details regarding channel formation in 3.3.2).
- **Additional channel width:** FPP channels will have $0.08 \times channel\ width$ of additional channel width per each side of the channel, resulting in a total of 0.016 of additional channel width. NPP best solution found was an additional channel width of $0.94 \times channel\ width$ per side, which means that it practically triples the original channel width. NonePP had $0.46 \times channel\ width$ of additional channel width per side (discussed in 3.3.2).
- **Resampling %:** The resampling values were not significantly different. Section 3.2 discusses this feature, for instance: a value of 0.08 for FPP means that each sample will have, at least, a 0.08 % difference from the previous sample closing price (discussed in 3.2.2).
- **Zig Zag filter:** NPP stands out, since it only considered peaks and troughs with a difference of, at least, 45 pips, while the remaining peaks and troughs were considered random price fluctuations (discussed in 3.3.1).
- **Pivot Points threshold:** FPP had a threshold of 0.82, which means that once the price has traveled a distance greater than 82% of the distance between the previous Pivot Point and the next Pivot Point in the direction of the price, then a position would not be opened as a precaution measure (discussed in 3.3.1). NPP completely ignored the use of Pivot Points, since its threshold was set to 99%. NonePP do not have a value assigned, since this strategy did not used Pivot Points.
- **ROI:** During the training phase, the solution with NonePP achieved the best ROI with a value of 26.5% in an year of training. Nevertheless, the results of the 3 strategies were very close to each other.

Quick Note: Although Resampling % and Zig Zag filter may seem very similar, the first one is used intends to homogenize the data set, by eliminating under and over sample periods, while the latter step in to filter random price fluctuations, enabling the system to identify significant peaks and troughs.

	Nº generations to converge	Nº candles	Additional channel width	Resampling %	Zig Zag filter (in pips)	Pivot Point threshold	ROI
Fibonnaci Pivot Points	26	2	0.08	0.08	3	0.82	23.5%
Normal Pivot Points	20	2	0.94	0.05	45	0.99	25.9%
No Pivot Points	28	2	0.46	0.04	13	---	26.5%

Table 4.3: Case Study A - Training Results.

The best parameters configuration for each strategy, found in the training phase, were applied in the testing phase. Figures 4.1, 4.2, 4.3 show, for the 3 compared strategies, the evolution of *ROI*, the price *variance* and the evolution of the price during the tested year (2018). In conjunction with the data from table 4.4, the following conclusions can be drawn:

- The FPP strategy, which had obtained the worst ROI during the training period, was the strategy that performed the best during the test period.

- The Sharpe Ratio of each strategy indicates that better risk-reward relation is achieved using the FPP strategy. It proves that trading with Fibonacci Pivot Points is better than trading with normal Pivot Points which in turn is always better than not using Pivot Points at all. This value is corroborated by the MDD value, where minimum fall in the accumulated ROI occurs using the FPP strategy. The difference in the Sharpe Ratio value between FPP and NPP may also be explained by the additional channel width in the NPP strategy, because a channel with such a large width becomes very insensitive to price variations that translate into having difficulty finding good exit (trading) points. Moreover, although NPP applies Pivot Points, it is tested using a 99% Pivot Point threshold, since it was the configuration of the best individual found by the GA.

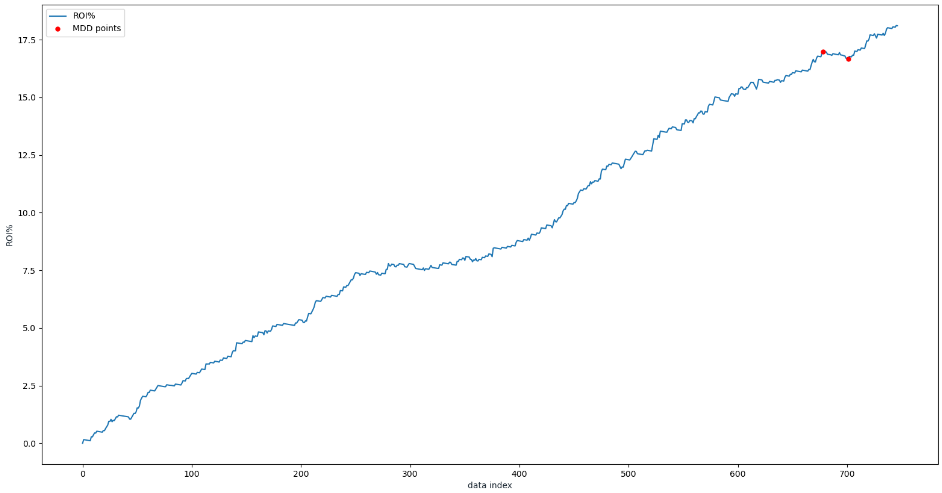
- It is interesting to verify that all strategies had a very similar rate of profitable trades, even more that the best strategy (FPP) is the one with the lowest rate of profitable trades. By inspecting the average profit per trade and the percentage of periods in the market, we can see that FPP was the strategy with highest profits per trade and where the proposed trading system was the least active on the market, only opened a position in 23.5% of all possible trades. NPP produces the highest rate of profitable trades, however each trade had much less profit than the FPP strategy, only summing 0.007% in ROI and was the most active on the market, taking 39% of all possible trades. The strategy with *No Pivot Points* (NonePP) stands out for having only 0.002% of average profit per trade.

- Since the spread and position size values are fixed, the amount of money spent in transactions increases linearly with the number of positions opened. Even though in a real scenario these values are variable, it is true that if we are able to reduce the number of times we open a position by having a higher average profit per trade, the ROI will increase and the transaction costs decrease.

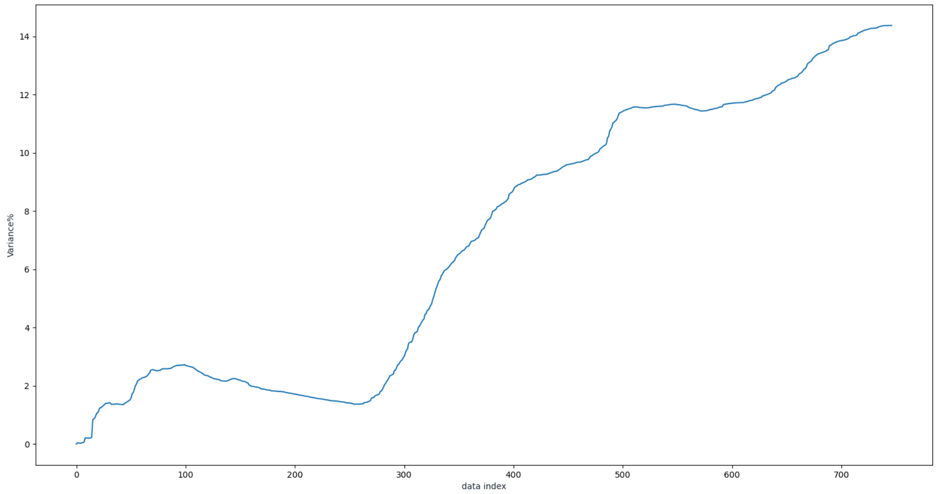
In summary, Fibonacci Pivot Points proved to be the best alternative to manage risk in the proposed strategy. This conclusion could have been reached just by inspecting ROI and Sharpe Ratio values. Just a final curiosity regarding the behaviour of these 3 strategies: in Figures 4.14.24.3, the reader can see that the FPP strategy's ROI has a much more stable and linear behaviour than the rest of the strategies. From Figure 4.1 it is also possible to infer that the strategy does not respond well to volatility increase, due to the fact that in the period where the variance increased, the ROI decreased.

	ROI	Sharpe Ratio	MDD (in accumulated ROI%)	Nº of open positions	Percentage of profitable trades	Avg. profit per trade (in ROI%)	Percentage of periods in the market	Transaction costs (in pips/dollars)
Fibonnaci Pivot Points	18.1%	3.85	- 0.31%	746	32.8%	0.024%	23.5%	746 pips / 7460 $
Normal Pivot Points	11.1%	2.02	- 1.34%	1649	37.2%	0.007%	39.0%	1649 pips / 16490 $
No Pivot Points	4.5%	0.28	- 1.80%	2810	34.2%	0.002%	33.6%	2810 pips / 28100 $

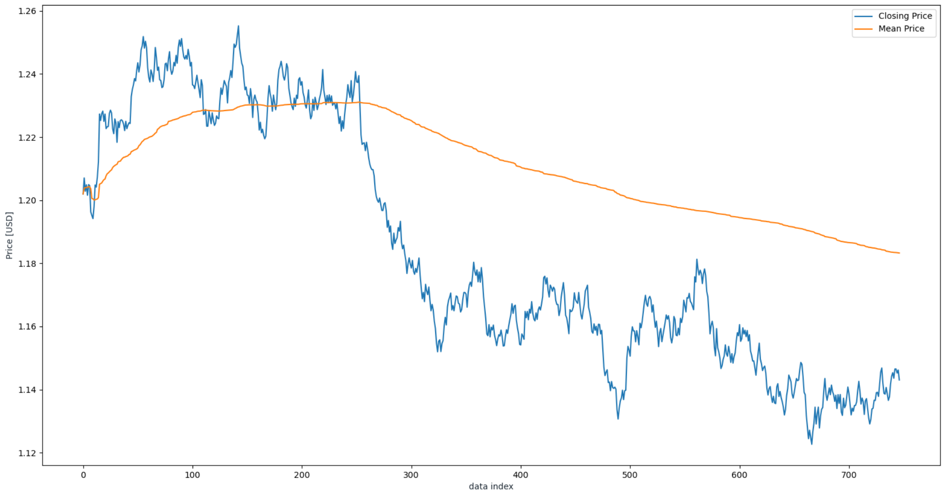
Table 4.4: Case Study A - Testing Results.



(a) ROI and MDD points.

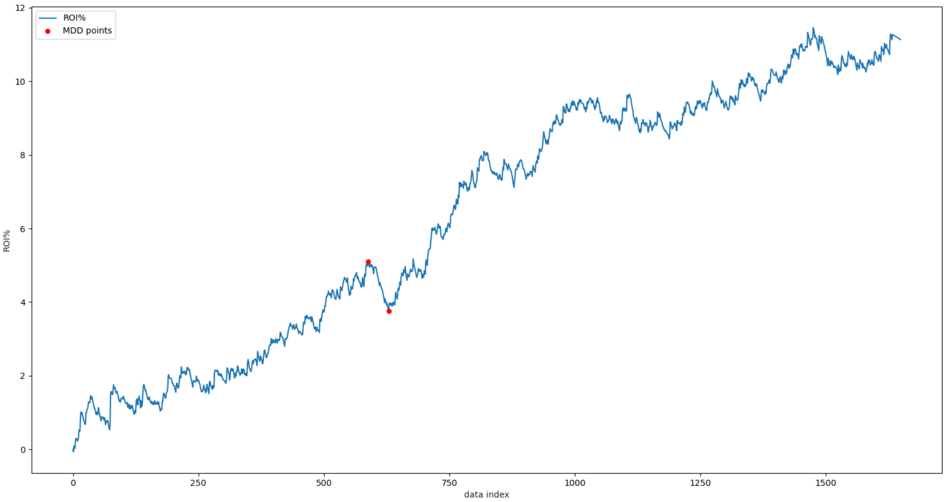


(b) Price Variance.

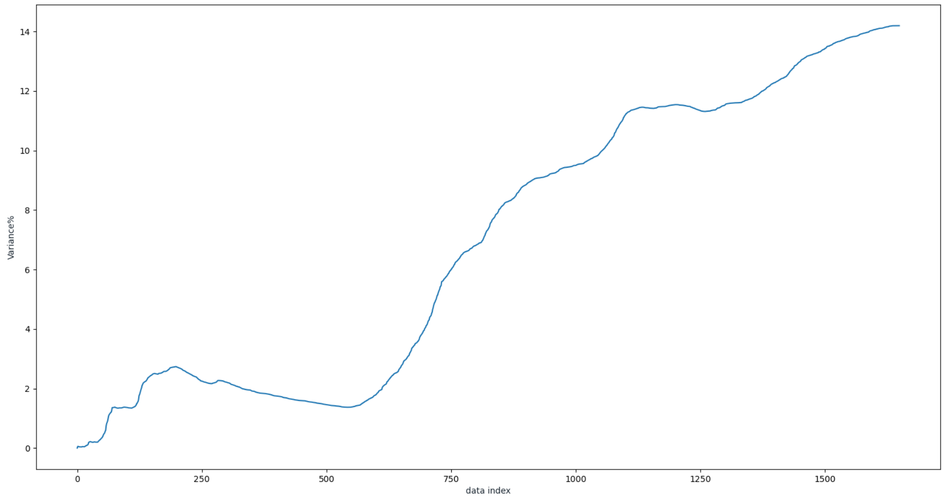


(c) Price evolution in comparison with mean price.

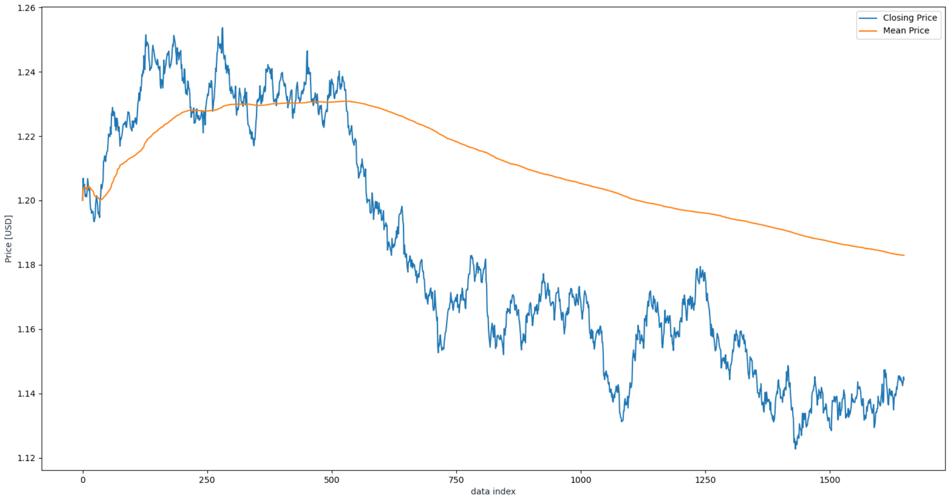
Figure 4.1: Fibonacci Pivot Points strategy.



(a) ROI and MDD points.

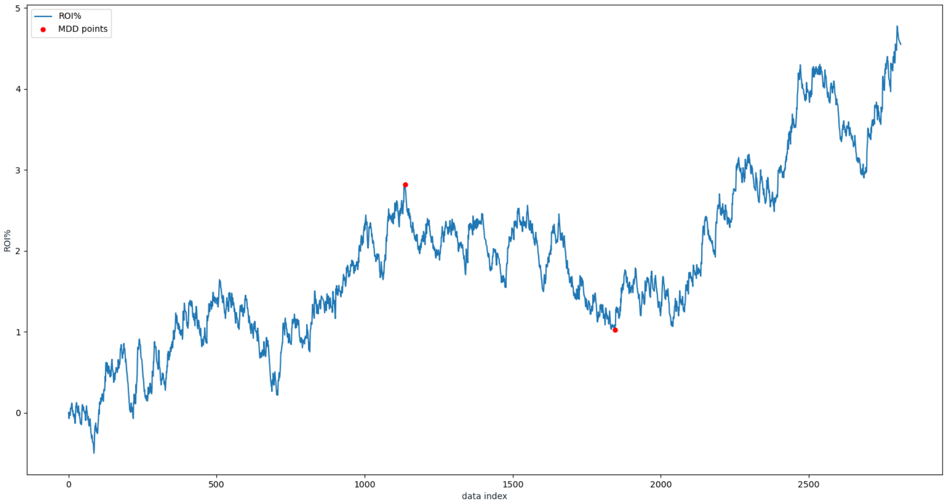


(b) Price Variance.

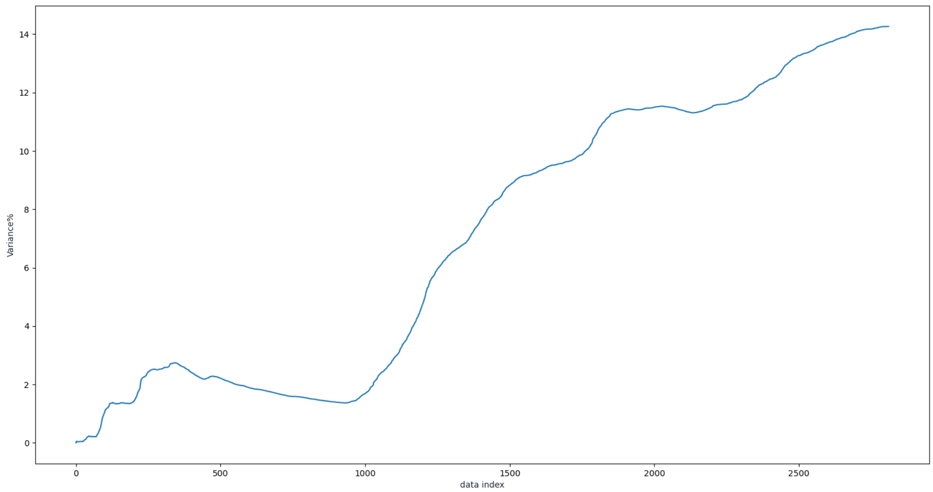


(c) Price evolution in comparison with mean price.

Figure 4.2: Normal Pivot Points strategy.



(a) ROI and MDD points.



(b) Price Variance.

(c) Price evolution in comparison with mean price.

Figure 4.3: No Pivot Points strategy.

4.2.2 Case Study B - Comparing different GA configuration setups

The goal of this case study is to compare the performance of the proposed trading strategy when varying some of the parameters that tune the GA (for a detailed explanation of each of the parameters read 2.1.4). As table 4.5 shows, there are 8 possible parameters that could be varied. Initially, only population size, generations maximum number and mutation rate were studied. Selecting population size is a sensitive issue; if the size of the population (search space) is small, this means little search space is available, and therefore it is possible to reach a local optimum. However, if the population size is very large, the area of search is increased and the computational load becomes high, therefore, the size of the population must be reasonable[33]. One of the parameters that compose the termination condition is the maximum number of generations ran by the GA, at the end of that specific number of generations it is assumed the algorithm has converged to an optimal solution. The other parameter that determines the termination of the GA is the maximum number of generations without evolution. In this work, a fixed value of 10 generations was set, due to the fact that 10 generations in a maximum of 50 corresponds to 20% of the total available time and 10% when the maximum number of generations is set to 100, which seems a reasonable value. Mutation rate is responsible for introducing diversity in the population, avoiding the GA from converging to local optimum solutions. That being said, combinations 1A to 1F, in table 4.5 represent different configurations only by varying these 3 parameters.

Then, 2 other combinations were tested: increasing crossover rate, in combination 2; decreasing elitism rate, in combination 3. If crossover rate is equal to 0%, the new generation of individuals is copied from the previous generation exactly as it is. As the crossover rate increases, the greater the mixture of genetic material between selected individuals, contributing a faster convergence and, hopefully, to better solutions. Elitism prevents the GA from losing its best individuals and in this case study 2 distinct values are tried: 10% (which is the one used in the original proposed system) and 2%. Excepting for the parameter tested, both combinations 2 and 3 have the remaining parameters configured with the values of the original proposed trading system.

Selection pressure and random immigrants rate were the other 2 parameters (beside maximum number of generations without evolving) that were not subject to evaluation. Selection pressure was set to k=3: k=2 introduces too much diversity, while k=4 already reduces it too much. Random immigrants rate was set to 20%, which in turn already generates a big amount of diversity, considering that the proposed solution has a population size of 50 individuals, resulting in 10 individuals with the worst fitness value being replaced by random individuals every generation. Nevertheless, although these were not considered the most relevant parameters in this case study, the 3 parameters that were not tested might provide valuable information and will be referenced for future works and improvements.

	Population Size	Generations max number	Mutation Rate	Crossover Rate	Elitism Rate	Selection Pressure	Max generation without evolving	Random Immigrants Rate
Combination 1A	50	100	10%	50%	10%	3	10	20%
Combination 1B	50	100	5%	50%	10%	3	10	20%
Combination 1C	50	50	5%	50%	10%	3	10	20%
Combination 1D	100	50	10%	50%	10%	3	10	20%
Combination 1E	100	100	5%	50%	10%	3	10	20%
Combination 1F	100	100	10%	50%	10%	3	10	20%
Combination 2	50	100	10%	80%	10%	3	10	20%
Combination 3	50	100	10%	50%	2%	3	10	20%

Table 4.5: Case Study B - Set of different GA configurations studied.

Table 4.7 shows the results of training the GA for the set of different combinations proposed, the training is done with 2017 EUR/USD currency pair data as can be seen in table 4.6 . At first sight, it is difficult to perceive the impact that each parameter might have in the ROI, therefore Pearson correlation [44] was applied. Pearson correlation is a measure of the strength of a linear association between two variables. Its range of values oscillate between -1 and +1, where 0 indicates that there is no association between the two variables, positive values show a positive linear correlation (i.e. when the value of one variable increases, the other increases as well) and negative values indicate a negative linear correlation (one value increases, the other decreases). That being said, the correlation between a certain parameter and the ROI was computed for: population size, generations maximum number and mutation rate. Crossover rate and elitism rate were not took into account because there is only one variation of combinations for each one of these two parameters and because it is easy to analyze the difference in ROI achieved just by inspecting the tables.
Regarding the training results, population size and generations maximum number showed a positive correlation with ROI of 0.40 and 0.42, respectively (as Figure 4.4 shows); mutation rate got a negative correlation with ROI of -0.06. Therefore, there is medium relation between the increase in ROI and population size, the same applies for generations maximum number. As for mutation rate, in the training phase at least, it had a practically null correlation with ROI.

By inspecting the training results for combination 2 (with changed elitism rate) and combination 1A (the original proposed solution), it can be seen that the only parameter that change between these two combinations is the elitism rate. An elitism rate of 10% in population of 50 individuals corresponds to 5 individuals passing directly to the next generation, while with an elitism rate of 2% only 1 individual (the best fitted individual of the population) passes directly to the next generation. Training results show that the ROI when using a smaller elitism rate (2%) were considerably lower than when using an elitism rate of 10%. It was expected that a lower elitism rate, by increasing diversity in the population, would lead to an increase in the number of generations to convergence, but that did not happen.
Combination 3 only differs from combination 1A in one parameter as well. In this case, the parameter

changed is the crossover rate. By increasing it to 80% a faster convergence was to be expected, as it happened. However, in exchange for a faster convergence, there is a significant decrease in ROI, 23.5% to 17.7%.

Simulation Parameters	Value
Market	Forex – EUR/USD
Training Period	2017
Test Period	2018
Position Size / Lot	100000 units / 1 Standard Lot
Spread	Fixed to 1 pip

Table 4.6: Case Study B - Simulation Parameters.

	Nº generations to converge	Nº candles	Additional channel width	Resampling %	Zig Zag filter (in pips)	Pivot Point threshold	ROI
Combination 1A	26	2	0.08	0.08	3	0.82	23.5%
Combination 1B	25	2	0.63	0.07	1	0.91	31.5%
Combination 1C	18	2	0.02	0.03	11	0.7	21.5%
Combination 1D	32	2	0.18	0.05	7	0.98	28.2%
Combination 1E	20	2	0.15	0.05	7	0.98	28.2%
Combination 1F	18	2	0.15	0.05	7	0.94	28.2%
Combination 2	20	2	0.11	0.03	12	0.99	18.4%
Combination 3	14	3	0.84	0.09	21	0.97	17.7%

Table 4.7: Case Study B - Training Results.

	Correlation coefficient	
Strength of Association	Positive	Negative
Small	[0.1, 0.3]	[-0.1, -0.3]
Medium	]0.3, 0.5]	]-0.3, -0.5]
Large	]0.5, 1.0]	]-0.5, -1.0]

(a) Pearson Correlation - strength of association.

	ROI
Population Size	0.40
Generations max number	0.42
Mutation Rate	-0.06

(b) Pearson Correlation between ROI and 3 parameters tested. association

Figure 4.4: Pearson Correlation - Training

In order to test these combinations, the parameters obtained for the best individuals for each combination, which may be found in the training results 4.7, were used to test the trading strategy for the

EUR/USD during 2018, as can be consulted in table 4.6.

The results obtained in the testing phase are described in table 4.8. Starting by analysing combination 2, where the elitism rate was decreased, it is possible to see that the ROI was practically nonexistent and the sharpe ratio got a negative value, which means the investment return is lower than the risk-free rate. Furthermore, among all the combinations, this was the one with the most erratic behaviour: the number of positions opened was 5116 and achieved the lowest percentage of profitable trades (with a value of 26.0%), ending up with an average profit per trade of approximately 0, which means this strategy produced a lot of false signals. Combination 3 performed slightly better, yet, when compared with combination 1A (since the only difference between these two is the value of crossover rate) the ROI and sharpe ratio were much lower. Combination 3 had a good percentage of profitable trades (39.5%), but the strategy is too conservative only being active in the market 19.7% of the the entire testing period.

Before, pearson correlation was ran against the training results. This time, pearson correlation included all the features of the GA being subject to analysis, as can be seen in table 4.9. Sharpe ratio and ROI present an almost perfect positive correlation, hence the correlation of one feature with ROI will be the same as with sharpe ratio. From the pearson correlation table and the prior analyis made, it can be concluded that:

- Population size has a big influence on the trading strategy returns. Between the values of 50 and 100, 100 should be the best choice for population size.
- Generations maximum number show a small influence in the final results, this can be due to the GA always converge before 50 generations because of the factor of the population being 10 generations without evolving. Being that the case, it does not matter if the maximum number of generations set is 50 or 100.
- Surprisingly, between 5% and 10%, mutation rate had a neglectable correlation coefficient with the ROI and sharpe ratio of the trading strategy results.
- Crossover rate exhibits a negative moderate correlation with ROI and Sharpe ratio, concluding that between a crossover rate of 50% or 80%, 50% is a better combination.
- Elitism rate shows a medium positive correlation with ROI and sharpe ratio, which indicates that a higher elitism rate would proportionate better results in the trading strategy. As seen in the testing results (table 4.8), the solution with an elitism rate of 10% (combination 1A) performed much better than combination 3, with 2% of elitism rate.
- Lastly, combination 1D was the one that performed the best in the testing phase, where its ROI almost didn't drop in relation to the ROI achieved during the training. It features a population size equal to 100, generations maximum number equal to 50, mutation rate of 10%, crossover rate of 50% and and elitism rate of 10%.

	ROI	Sharpe Ratio	MDD (in accumulated ROI%)	Nº of open positions	Percentage of profitable trades	Avg. profit per trade (in ROI%)	Percentage of periods in the market	Transaction costs (in pips/dollars)
Combination 1A	18.1%	3.85	- 0.31%	746	32.8%	0.024%	23.5%	746 pips / 7460 $
Combination 1B	23.1%	5.17	- 0.91%	1278	41.5%	0.002%	43.3%	1278 pips / 12780 $
Combination 1C	-3.1%	-1.74	- 3.20%	1703	18.5%	-0.002%	7.1%	1703 pips / 17030 $
Combination 1D	27.4%	6.33	- 0.61%	2333	34.7%	0.012%	30.9%	2333 pips / 23330 $
Combination 1E	25.8%	5.93	- 0.76%	2383	33.2%	0.011%	30.0%	2383 pips / 23830 $
Combination 1F	25.9%	5.94	- 0.76%	2382	33.2%	0.011%	30.0%	2382 pips / 23820 $
Combination 2	0.2%	-0.86	- 2.69%	5116	26.0%	$4.7e^{-5}$	24.9%	5116 pips / 51160 $
Combination 3	5.5%	0.55	- 1.00%	306	39.5%	0.018%	19.7%	306 pips / 3060 $

Table 4.8: Case Study B - Testing Results.

	ROI	Sharpe Ratio
Population Size	0.73	0.73
Generations max number	0.15	0.15
Mutation Rate	0.06	0.06
Crossover Rate	-0.49	-0.49
Elitism Rate	0.32	0.32
Sharpe Ratio	0.99	1

Table 4.9: Case Study B - Pearson correlation for testing results.

4.2.3 Case Study C - Comparing the trading strategy for major, minor and exotic currency pairs

This case study aims to compare the proposed trading strategy's performance for one major, one minor and one exotic currency pair.

Major currency pairs are the most traded currency pairs, usually including the USD currency. Due to its high volume, these pairs offer a large amount of liquidity and a fairly amount of volatility, while offering price stability. Having more traders in the market, the brokers are able to provide low spreads to trade major currency pairs. Minor currency pairs are pairs that are less traded than majors and normally do not include USD currency but include other major currency. They offer a moderate liquidity, hence their spreads tend to be higher than the ones provided for trading majors. Also, their volatility is moderate as well. Finally, exotic currency pairs use currencies that are thinly traded in Forex market, usually belonging to emergent market economies. Exotic currency pairs normally include one major currency (like USD) one currency of a developing economy (like ZAR, from South Africa). These pairs are highly illiquid and, since there are not enough buyers and sellers that trade in these markets, have very high spreads. Due to its lack of volume along with unstable economic conditions, exotic currency pairs can be extremely volatile.

The currency pairs elected to conduct this case study where:

- **Major currency pair:** EUR/USD.
- **Minor currency pair:** EUR/JPY.
- **Exotic currency pair:** USD/NOK.

The currency pair EUR/USD is the pair that was used to develop this thesis's proposed solution, therefore it makes sense that it should be a benchmark for the minor and exotic currency pairs. In order to have a term of comparison between each currency pair performance, EUR/JPY and USD/NOK were chosen to conduct this study, because they have a very high correlation value with EUR/USD and between themselves. This means that when the value of one pair increases the other increases too, in the case of a positive correlation; when the value of one pair increases the other decreases, in the case of a negative correlation. If they were not highly correlated, it would become difficult to compare their performances, because, for instance: it could be a great year for EUR/JPY with the market always trending in a stable manner and the EUR/USD market could have been sideways during this entire period. In the end, we would conclude that EUR/JPY was more suited for this strategy, when it might not be the case. Again, this problem is mitigated by choosing currency pairs highly correlated, as Figure 4.5 shows.

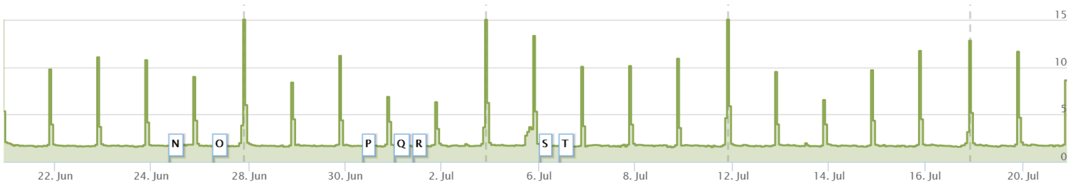
	EURUSD	EURJPY	USDNOK
EURUSD		90	-93
EURJPY	90		-95
USDNOK	-93	-95	

Figure 4.5: Case Study C - Currency pairs correlations (adapted from: https://www.mataf.net/en/forex/tools/correlation).

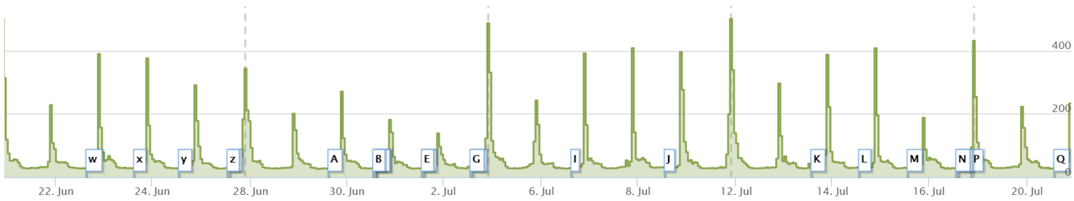
The GA trained in the year of 2017 for each of the currency pairs and was tested in the year of 2018, as table 4.10 shows. In order to define a fixed spread for EUR/JPY and USD/NOK pairs (EUR/USD already used a 1 pip fixed spread), a visual analysis of spreads values, for each of the currency pairs, during 1 month period was made. According to 4.6, the spread values chosen were: 2 pips for EUR/JPY, 50 pips for USD/NOK.

Simulation Parameters	Value
Market	Forex – EUR/USD, EUR/JPY, USD/NOK
Training Period	2017
Test Period	2018
Position Size / Lot	100000 units / 1 Standard Lot
Spread	1 pip, 2 pips, 50 pips

Table 4.10: Case Study C - Simulation Parameters.



(a) EUR/JPY spread.



(b) USD/NOK spread.

Figure 4.6: Spread values - 1 month data (adapted from: https://www1.oanda.com/lang/pt/forex-trading/markets/recent).

From the GA training results, presented in table 4.11, some values stand out:

- EUR/JPY: Huge value of ROI when compared with the other two currency pairs. The testing results will confirm if this value derives from overfitting or not. Due to a zig zag filter value equal to 1, all possible peaks and troughs will be considered to build a channel, which translates in a high number of opened positions in the testing data.
- USD/NOK: A very high resampling value, meaning that the testing dataset will have few samples after resampling. The resampling value combined with the illiquidity of USD/NOK and the high value of the zig zag filter might results in a low number of opened positions in the testing results, when compared with the other two pairs.

	Nº generations to converge	Nº candles	Additional channel width	Resampling %	Zig Zag filter (in pips)	Pivot Point threshold	ROI
EUR/USD	26	2	0.08	0.08	3	0.82	23.5%
EUR/JPY	24	2	0.03	0.02	1	0.98	96.0%
USD/NOK	43	2	0.24	0.29	46	0.86	21.6%

Table 4.11: Case Study C - Training Results.

The best parameters configuration for each currency pair, found in the training phase, were applied in the testing phase. In conjunction with the data from table 4.12, the following conclusions can be drawn:

- EUR/JPY had a ROI more than 4 times the ROI with EUR/USD. USD/NOK presents a low ROI. The high ROI showed that the training results obtained for EUR/JPY were not a result of overfitting.
- Interestingly, despite the high value of EUR/JPY's ROI, EUR/USD showed a far superior sharpe ratio. Both EUR/JPY and USD/NOK got very low values, which could be explained by their moderate and high volatility (respectively), meaning that EUR/USD would be a better choice concerning risk-adjusted return.
- Due to its minimum zig zag filter value and low resampling %, the EUR/JPY strategy opened 15045 positions, with a win rate of only 21.7% and being the most active strategy in the market (among the three tested) with 31.1% of time spent trading. On the other hand, the trading system using USD/NOK data was much more conservative, only opening 98 positions. In this case the opposite happened, a high resampling % combined with a high zig zag filter resulted in a data set with few points, which leads to less channels built and hence less opened positions. Nonetheless, those characteristics allowed this strategy to achieve a win rate of 42.9%.
- As stated before, admitting a fixed spread, it is possible to compare the difference in transaction cost between an exotic currency pair and a major/minor currency pair. Surprisingly, the USD/NOK was able to achieve a positive ROI, given the low amount of opened positions. The USD/NOK strategy did not face major drawdowns, as can be seen in Figure 4.7, probably due to its conservative characteristics, otherwise, being a volatile market, it would receive more false trading signals.

The proposed trading system demonstrated capability of being profitable when trading with any of the currency pairs studied. However, given the immense difference between their sharpe ratio values, it can be concluded that a major currency pair would be more suitable to trade with this system.

	ROI	Sharpe Ratio	MDD (in accumulated ROI%)	Nº of open positions	Percentage of profitable trades	Avg. profit per trade (in ROI%)	Percentage of periods in the market	Transaction costs (in pips/dollars)
EUR/USD	18.1%	3.85	- 0.31%	746	32.8%	0.024%	23.5%	746 pips / 7460 $
EUR/JPY	73.5%	0.28	- 0.22%	15045	21.7%	0.005%	31.1%	15045 pips / 300900 $
USD/NOK	7.7%	0.15	- 1.42%	98	42.9%	0.079%	21.7%	98 pips / 49000 $

Table 4.12: Case Study C - Testing Results.

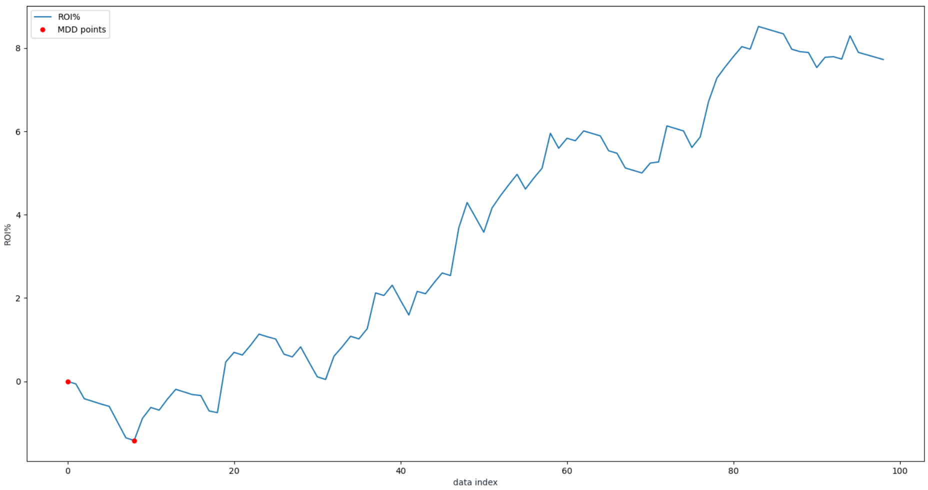


Figure 4.7: USD/NOK - ROI.

4.2.4 Case Study D - Best Results for EUR/USD

Throughout the previous case studies the goal was to compare the performance of the trading strategy when subject to different configurations (case studies A & B) and working in different environments(case study C), therefore only 1 year of data was used to train the GA. This case study serves the purpose of demonstrating the best results achieved by this system when trained with 2 years of data, using the original GA parameters, shown in table 4.13.The trading strategy was trained in the year of 2015 and 2017 and tested, individually, during the years of 2017, 2018, 2019 and 2020, using fixed spread and position size, as can be seen in table 4.14

GA Parameters	Value
Population Size	50
Elitism Rate	10%
Tournament Selection – Selection Pressure	k = 3
Mutation Rate	10%
Random Immigrants Rate	20%
Crossover Rate	50%
Termination Criteria	100 generations or 10 generations without evolution

Table 4.13: Case Study D - GA Parameters.

Simulation Parameters	Value
Market	Forex – EUR/USD
Training Period	[2015, 2016]
Test Period	2017, 2018, 2019, 2020
Position Size / Lot	100000 units / 1 Standard Lot
Spread	Fixed to 1 pip

Table 4.14: Case Study D - Simulation Parameters.

Training the GA with 2 years of data resulted in a ROI of 78.7%, which translates into 39.35% annualized ROI (table 4.15). Compared with the training results obtained for 2017 (table 4.16), there is a difference of 16.35% in ROI obtained. Although this comparison is made for different years, the testing results confirmed the difference in performance.

	Nº generations to converge	Nº candles	Additional channel width	Resampling %	Zig Zag filter (in pips)	Pivot Point threshold	ROI
2015/2016	30	2	0.27	0.11	3	0.9	78.7%

Table 4.15: Case Study D - Training Results.

	Nº generations to converge	Nº candles	Additional channel width	Resampling %	Zig Zag filter (in pips)	Pivot Point threshold	ROI
2017	26	2	0.08	0.08	3	0.82	23.5%

Table 4.16: Case Study D - 2017 Training Results.

The testing results are presented in table 4.17. Compared with the results obtained for the year of 2018 (table 4.18), which was trained with 2017 data, it is possible to see that:

- Excepting for 2019, the ROI was consistently higher when using 2 years of training. The best result was achieved in 2020, obtaining a ROI of 34.9%, as Figure 4.8 illustrates.
- As expected, since EUR/USD is a major currency and does not present a lot of volatility, the sharpe ratio values were good (superior to 1) and even in 2019, where ROI was lower, the sharpe ratio reached the highest value among all years tested.
- Win rate rised in comparison with the one year trained training strategy, which means the strategy increased its capability in predicting price movements.
- Comparing directly the results for 2018, the trading strategy trained with 2 years of data achieved a ROI and sharpe ratio of 32.5% and 7.71, respectively; while the one year trained trading strategy achieved a ROI and sharpe ratio of 18.1% and 3.85%, respectively.

This case study shows the performance of the proposed trading strategy using 2 years of data to train the GA. It showed that it outperformed the results obtained with one year of training. In future developments as well as in a real-life scenario, a greater amount of data should be used to optimize this trading strategy, in order to reach its maximum performance.

	ROI	Sharpe Ratio	MDD (in accumulated ROI%)	Nº of open positions	Percentage of profitable trades	Avg. profit per trade (in ROI%)	Percentage of periods in the market	Transaction costs (in pips/dollars)
2017	23.1%	3.66	- 0.92%	552	39.3%	0.042%	36.8%	552 pips / 5520 $
2018	32.5%	7.71	- 0.66%	638	45.9%	0.051%	39.4%	638 pips / 6380 $
2019	17.0%	9.35	- 0.56%	304	46.4%	0.056%	38.3%	304 pips / 3040 $
2020	34.9%	7.22	- 0.77%	784	42.1%	0.044%	38.4%	784 pips / 7840 $

Table 4.17: Case Study D - Testing Results.

	ROI	Sharpe Ratio	MDD (in accumulated ROI%)	Nº of open positions	Percentage of profitable trades	Avg. profit per trade (in ROI%)	Percentage of periods in the market	Transaction costs (in pips/dollars)
2018	18.1%	3.85	- 0.31%	746	32.8%	0.024%	23.5%	746 pips / 7460 $

Table 4.18: Case Study D - 2018 Testing Results (Trained with 2017 data).

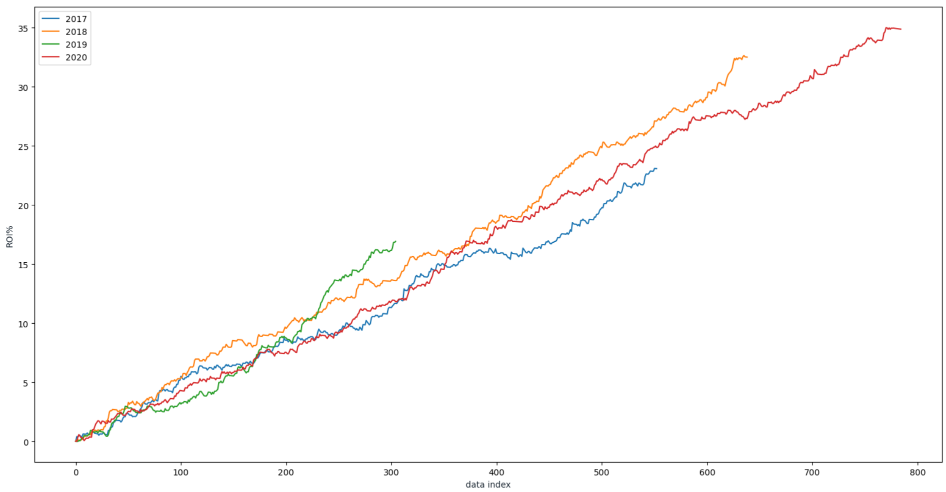


Figure 4.8: Case Study D - ROI of the tested years.

4.3 Chapter Conclusions

This chapter presented the tests performed to the proposed trading strategy and presented its performance.

In case study A (4.2.1), 3 different approaches of the strategy were tested. These approaches concern the application of the technical indicator pivot points, whose purpose is to measure the risk of a possible trade. The results showed that, as original proposed, fibonacci pivot points were the best choice for this solution.

In case study B (4.2.2), different combinations of the GA parameters were tested. GA's population size turned out to be the most influential for the performance of the trading system and combination D was the one that performed the best.

Case study C (4.2.3) studied the performance of the system when trading in different environments, i.e. with different currency pairs, with one pair belonging to the group of major currency pairs, the second to the minor currency pairs and the last was an exotic pair. The 3 currency pairs elected produced a positive ROI, however the major and minor pairs have reached better results. The major pair, EUR/USD, was concluded to be the best choice to apply in this trading system, due to its sharpe ratio being much higher than the remaining.

In case study D (4.2.4) the performance of the system, using 2 years (2015 & 2016) of data to train the GA and 4 years to test the solution ([2017,2020]), was displayed, being that the best ROI was achieved in 2020 with a value of 34.9%.

Chapter 5

Conclusions

5.1 Conclusions

This work implements a solution to trade Foreign Exchange Market currency pairs in trending markets, which means it always trades in the direction of the trend and if there is no trend it does not open a position. In order to build this system, two essential points were combined: Technical Analysis and Genetic Algorithms. Technical analysis provided the technical indicators used to create the investment strategy, which dictates the moments of market entry and exit. The technical indicators used where channel pattern (belongs to chart analysis, but is included in TA) and fibonacci pivot points. GA optimized the parameters involved in the investment strategy, without it the parameters values employed would be arbitrary and the best possible performance would not be achieved. Achieved results showed the proposed system is better suited to trade major currency pairs, namely EUR/USD. Furthermore, the results showed that a better performance is achieved when training the GA with a bigger population size. The proposed system was able to reach 34.9% ROI in 2020 test data, training the GA with 2015 and 2016 data. It is noteworthy that a trading strategy based on channeling is clearly uncommon; in fact, in the literature only one work was found that approached this thematic and was not able to achieve positive returns while applying channel trading.

5.2 Future Work

Despite having accomplished this work's objectives, some limitations that became noticeable during the development of this thesis and some improvements which may improve the solution's performance are suggested:

- Develop a mechanics to ingest data in nearly real-time and being continuously evaluating the market. This would add more complexity to the model, being subject to a constant processing effort. A dedicated storage unit should also be considered with a retention data policy, allowing data not be immediately lost after running the algorithm in case of an unexpected error occurs during the process. Besides it allows to identify more easily miss ingested files. Ideally, this would be done recurring to a solution in the cloud, where the resources may be allocated and deallocated according to our needs. The model would need to be re-trained once in a while, in the presented solution it is trained on-demand and for a fixed period, but considering that it would be re-trained periodically a rolling window method would be the most appropriate.

- As attempted initially, use ADX or other technical indicator that return the strength of a trend, enabling the system to only enter strong trends i.e. with high chances of profit. In this solution, it is done by considering only channels where a candle's price is consecutively higher (in an up-trend)/lower (in a downtrend) than the previous candle for x times, being x the number of candles needed to form a channel (this parameter is optimized by the GA).

- Improve the solution by adding variable position size, leverage and spread, in order to replicate the exact conditions of the market during the simulation. At the moment these parameters are set to a fixed value.

- Increase the training data set. The proposed system was trained for one and two years, the performance increased while training the model with two year of data. The longer the training time, the more different market conditions the model knows and the better predictive capacity it will have.

- To optimize the GA (as discussed in 4.2.2) explore different combinations varying selection pressure and random immigrants rate: these parameters may increase the population diversity, which in turn might contribute to achieve solutions with a better return (which is, after all, the goal of an investment strategy).

- Change from a static to a dynamic GA, for instance applying hyper-mutation, hyper-selection or associative memory; allowing the GA to adapt to different environments.

Bibliography

[1] R. C. et al. Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications*, 2016. doi:10.1016/j.eswa.2016.02.006.

[2] J. M. B. Carapuço. Reinforcement learning applied to forex trading, Nov 2017. URL `https://fenix.tecnico.ulisboa.pt/cursos/meft/dissertacao/1409728525632011`.

[3] A. Hayes. Pip definition. `https://www.investopedia.com/terms/p/pip.asp`, .

[4] A. Hayes. Position. `https://www.investopedia.com/terms/p/position.asp`, .

[5] N. Lioudis. Forex leverage: A doubled-edged sword. `https://www.investopedia.com/articles/forex/07/forex_leverage.asp`.

[6] S. B. Achelis. *Technical Analysis from A to Z*. McGraw Hill New York, 2^{nd} edition, 2001. ISBN:978-0-07-138011-9.

[7] J. J. Murphy. *TECHNICAL ANALYSIS OF THE FINANCIAL MARKETS*. New York Institute of Finance, 2^{nd} edition, 1999. ISBN:0-7352-066-1.

[8] C. D. Kirkpatrick and J. R. Dahlquist. *Technical Analysis: The Complete Resource for Financial Market Technicians*. FT Press, 2^{nd} edition, 2011. ISBN:10: 0-13-705944-2.

[9] T. Segal. Fundamental analysis. `https://www.investopedia.com/terms/f/fundamentalanalysis.asp`.

[10] S. Jakpar, M. Tinggi, A. H. Tak, and C. W. Yi. Fundamental analysis vs technical analysis: The comparison of two analysis in malaysia stock market. *UNIMAS Review of Accounting and Finance*, 2018. doi:10.33736/uraf.1208.2018.

[11] J. L. Bettman, S. J. Sault, and E. L. Schultz. Fundamental and technical analysis: substitutes or complements? *Accounting and Finance*, 2009. doi:10.1111/j.1467-629x.2008.00277.x.

[12] B. M. NWIDOBIE. The random walk theory: An empirical test in the nigerian capital market. *Asian Economic and Financial Review*, 2014.

[13] B. G. Malkiel. The efficient market hypothesis and its critics. *Journal of Economic Perspectives*, 2003. doi:10.1257/089533003321164958.

[14] S. DUPERNEX. Why might share prices follow a random walk? *Student Economic Review*, 2007.

[15] Y. S. Abu-Mostafa and A. F. Atiya. Introduction to financial forecasting. *Kluwer Academic Publishers*, 1996.

[16] C. J. Neely. Technical analysis in the foreign exchange market: A layman's guide. *Federal Reserve Bank of St. Louis*, 1997. doi:10.20955/r.79.23-38.

[17] L. A. Teixeira and A. L. Oliveira. A method for automatic stock trading combining technical analysis and nearest neighbor classification. *Expert Systems with Applications*, 2010. doi:10.1016/j.eswa.2010.03.033.

[18] P. Fernández-Blanco, D. J. B. Sagi, F. Soltero, and I. Hidalgo. Technical market indicators optimization using evolutionary algorithms. *Genetic and Evolutionary Computation Conference, GECCO 2008*, 2008. doi:10.1145/1388969.1388989.

[19] Y. Hua, K. Liub, X. Zhanga, L. Sub, E. Ngai, and M. Liu. Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review. *Applied Soft Computing*, 2015. https://doi.org/10.1016/j.asoc.2015.07.008.

[20] J. Chen. Zig zag indicator. `https://www.investopedia.com/terms/z/zig_zag_indicator.asp`.

[21] C. L. Person. *A Complete Guide to Technical Trading Tactics: How to Profit Using Pivot Points, Candlesticks Other Indicators*. John Wiley Sons, 2004.

[22] T. Bartz-Beielstein, J. Branke, J. Mehnen, and O. Mersmann. Evolutionary algorithms. *WIREs Data Mining Knowl Discov*, 2014. doi:10.1002/widm.1124.

[23] A. Shukla, H. M. Pandey, and D. Mehrotra. Comparative review of selection techniques in genetic algorithm. *IEEE*, 2015. doi:10.1109/ABLAZE.2015.7154916.

[24] N. M. Razali and J. Geraghty. Genetic algorithm performance with different selection strategies in solving tsp. *Newswood Limited*, 2011. ISBN: 978-988-19251-4-5.

[25] T. Blickle and L. Thiele. A comparison of selection schemes used in genetic algorithms. *MIT Press*, 1996. doi:10.1162/evco.1996.4.4.361.

[26] A. Umbarkar and P. Sheth. Crossover operators in genetic algorithms: A review. *ICTACT JOURNAL ON SOFT COMPUTING*, 2015. doi:10.21917/ijsc.2015.0150.

[27] R. Poli and W. Langdon. On the search properties of different crossover operators in genetic programming. 2001.

[28] M. A. H. Dempster and C. M. Jones. Can channel pattern trading be profitably automated? *The European Journal of Finance*, 2010. doi:10.1080/13518470110052831.

[29] Y.-H. Lui and D. Mole. The use of fundamental and technical analyses by foreign exchange dealers: Hong kong evidence. *Journal of International Money and Finance*, 1998. https://doi.org/10.1016/S0261-5606(98)00011-4.

[30] I. Moosa and L. Li. Technical and fundamental trading in the chinese stock market: Evidence based on time-series and panel data. *Emerging Markets Finance and Trade*, 2011. doi:10.2753/REE1540-496X4701S103.

[31] A. S. Wafi, H. Hassan, and A. Mabrouk. Fundamental analysis vs technical analysis in the egyptian stock exchange – empirical study. *IJBMS*, 2015.

[32] J. Coakley, M. Marzano, and J. Nankervis. How profitable are fx technical trading rules? *IEEE*, 2016. http://dx.doi.org/10.1016/j.irfa.2016.03.010.

[33] A. H. et al. Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. *MDPI*, 2019. doi:10.3390/info10120390.

[34] A. Elshamli, H. A. Abdullah, and S. Areibi. Genetic algorithm for dynamic path planning. *IEEE*, 2004. doi:10.1109/CCECE.2004.1345203.

[35] C. Evans, K. Pappas, and F. Xhafa. Utilizing artificial neural networks and genetic algorithms to build an algo-trading model for intra-day foreign exchange speculation. *Mathematical and Computer Modelling*, 2013. https://doi.org/10.1016/j.mcm.2013.02.002.

[36] C. Aranha and H. Iba. Modelling cost into a genetic algorithm-based portfolio optimization system by seeding and objective sharing. *IEEE*, 2007. doi:10.1109/CEC.2007.4424472.

[37] A. Hirabayashi, C. Aranha, and H. Iba. Optimization of the trading rule in foreign exchange using genetic algorithm. *Genetic and Evolutionary Computation Conference, GECCO 2009, Proceedings, Montreal, Québec, Canada*, 2009. doi:10.1145/1569901.1570106.

[38] P. B. Myszkowski and A. Bicz. Evolutionary algorithm in forex trade strategy generation. *Proceedings of the International Multiconference on Computer Science and Information Technology pp. 81–88*, 2010. doi:10.1109/IMCSIT.2010.5679921.

[39] H. Zhang and R. Ren. High frequency foreign exchange trading strategies based on genetic algorithms. *IEEE*, 2010. doi:10.1109/NSWCTC.2010.234.

[40] S. Deng and A. Sakurai. Foreign exchange trading rules using a single technical indicator from multiple timeframes. *IEEE*, 2013. doi:10.1109/WAINA.2013.7.

[41] Z. Zhang and M. Khushi. Ga-mssr: Genetic algorithm maximizing sharpe and sterling ratio method for robotrading. *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020. doi:10.1109/IJCNN48605.2020.9206647.

[42] M. L. D. PRADO. *Advances in Financial Machine Learning*. y John Wiley Sons, 2018. ISBN:978-1-119-48211-6.

[43] J. Fernando. Return on investment (roi). `https://www.investopedia.com/terms/r/returnoninvestment.asp`.

[44] Pearson product-moment correlation. https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php.