Financial forecasting with the combination of Physical and Event-based time using Genetic Programming

Xinpeng Long

A thesis submitted for the degree of

Doctor of Philosophy

at the School of Computer Science and Electronic Engineering

University of Essex

October 4, 2024

Acknowledgements

First of all, I would like to express my deepest gratitude to my supervisor, Dr Michael Kampouridis, for his patient support, encouragement and guidance throughout my whole PhD. His insight and expertise were invaluable to the completion of this thesis. Second, I would like to extend my appreciation to Dr Delaram Jarchi, Dr Panagiotis Kanellopoulos, and Dr Tasos Papastylianou for their thoughtful feedback and suggestions, which have significantly enhanced the quality of my research papers. I would also like to thank my friends, who provided encouragement through challenging times when continuing seemed impossible. Finally, I am deeply grateful to my family for providing financial support and emotional encouragement, allowing me to pursue my research without additional worries. I sincerely appreciate everyone who has contributed to this valuable and unforgettable journey, whether mentioned here or not. Your support has made this accomplishment possible.

Abstract

This thesis explores the application of genetic programming (GP) within the directional changes (DC) framework for algorithmic trading. Traditional algorithmic trading methods rely on datasets with fixed time intervals, such as hourly or daily data, leading to a discontinuous representation of time. DC provides an alternative by transforming these datasets into event-driven sequences, allowing for a unique price analysis approach. The first part of the thesis compares GP with machine learning (ML) algorithms in algorithmic trading, focusing on factors like market data, time periods, forecasting windows, and transaction costs—variables often neglected in previous studies. A comprehensive evaluation of a GP-based financial approach is conducted, comparing it to nine popular ML algorithms and the buy-and-hold strategy, using daily data from 220 datasets across 10 international markets. Results show that GP not only yields profitable results but also outperforms ML algorithms in terms of risk and Sharpe ratio. The second part investigates GP within the DC framework, introducing two novel algorithms: GP-DC, which uses only DC-based indicators, and GP-DC-PT, which combines DC-based and physical-time indicators from technical analysis. Both approaches outperform non-DC-based GP strategies, technical analysis, and buy-and-hold benchmarks, with GP-DC-PT achieving an average return of over 18%, highlighting the advantage of incorporating DC into trading strategies. Finally, the thesis introduces two multi-objective optimization algorithms, MOO2 and MOO3, based on the NSGA-II framework, which optimize two and three fitness functions, respectively, using DC and physical-time indicators. Both MOO2 and MOO3 outperform single-objective methods, with MOO3 showing consistent improvements across all metrics. These findings suggest that incorporating directional changes significantly enhances trading strategies' return and risk performance.

List of Publications

- Long, Xinpeng, Michael Kampouridis, and Delaram Jarchi. "An in-depth investigation of genetic programming and nine other machine learning algorithms in a financial forecasting problem." 2022 IEEE Congress on Evolutionary Computation (CEC).
- Long, Xinpeng, Michael Kampouridis, and Panagiotis Kanellopoulos. "Genetic programming for combining directional changes indicators in international stock markets." 2022 International Conference on Parallel Problem Solving from Nature (PPSN).
- Long, Xinpeng, Michael Kampouridis, and Panagiotis Kanellopoulos. "Multi-objective optimisation and genetic programming for trading by combining directional changes and technical indicators." 2023 IEEE Congress on Evolutionary Computation (CEC).
- Long, Xinpeng and Michael Kampouridis. "α-dominance two-objective Optimization Genetic Programming for Algorithmic Trading under a Directional Changes Environment", 2024 IEEE Congress on Evolutionary Computation (CIFEr).

List of Works Under Review

- Long, Xinpeng, Michael Kampouridis, and Panagiotis Kanellopoulos. "An in-depth investigation of genetic programming under physical time and directional change frameworks for algorithmic trading", International Journal of Intelligent Systems.
- Long, Xinpeng, Michael Kampouridis, and Tasos Papastylianou. "Three-objective Genetic Programming approach for algorithmic trading under a Directional Changes Environment", Expert Systems with Applications.

Contents

1 Introduction **1**

1.1 Motivation 2

1.2 Thesis overview 5

2 Background **6**

2.1 Financial forecasting 6

2.1.1 Physical time series prediction 7

2.1.2 Physical time series indicators 8

2.1.3 Event-based prediction 11

2.1.4 Directional changes indicators 13

2.2 Genetic programming 16

2.2.1 General information 16

2.2.2 Genetic programming representations 17

2.2.3 Population Initialisation 18

2.2.4 Genetic Operators 20

2.2.5 Breeding method 22

2.2.6 Selection 23

2.3 Multi-objective optimisation 24

2.3.1 NSGA-II algorithm 24

2.3.1.1 Fast non-dominated sorting approach 25

2.3.1.2 Crowding distance 27

2.3.1.3 Competition of parents and offspring 29

2.3.2 α-dominance strategy 29
2.4 Conclusion 31

3 Literature review **32**
3.1 Physical time series prediction 32
3.1.1 Gradient boosted machine 33
3.1.2 Support vector machine 34
3.1.3 Multi-layer perceptron 35
3.1.4 K-nearest neighbours 35
3.1.5 Decision tree 36
3.1.6 Random forest 37
3.1.7 Extra trees 37
3.2 Genetic programming (GP) 38
3.3 Directional changes 40
3.4 Multi-objective optimisation 41
3.5 Conclusion 43

4 Genetic programming application on physical time scale **45**
4.1 Motivation 45
4.2 Methodology 46
4.2.1 Genetic programming model 47
4.2.1.1 Terminal set 47
4.2.1.2 Function set 47
4.2.1.3 Model representation 47
4.2.1.4 Selection method 49
4.2.2 Trading strategy 50
4.2.3 Fitness function 51
4.3 Experimental set-up 52
4.3.1 Data 52
4.3.2 Benchmarks 53

4.3.3 Parameter tuning . . . 54
4.4 Results and analysis . . . 54
4.4.1 GP vs ML algorithms . . . 55
4.4.2 Market and countries . . . 56
4.4.3 Periods . . . 58
4.4.4 Buy-and-hold strategy . . . 58
4.5 Conclusion . . . 59

5 GP application to an event-based framework 61
5.1 Motivation . . . 61
5.2 Methodology . . . 62
5.2.1 Directional changes and technical indicators . . . 62
5.2.2 Model representation . . . 64
5.2.3 Remainder of the GP configuration . . . 64
5.3 Experimental set-up . . . 65
5.3.1 Benchmarks . . . 65
5.3.2 Parameter tuning . . . 66
5.4 Results and analysis . . . 66
5.4.1 Comparison of the GP-based algorithms . . . 67
5.4.1.1 Periods . . . 67
5.4.1.2 Summary . . . 70
5.4.2 Comparison of GP-DC-PT algorithm to the nine ML algorithms . . . 71
5.4.3 Comparison of GP-DC-PT algorithm to other technical indicators . . . 77
5.4.4 Buy-and-hold strategy . . . 78
5.5 Conclusion . . . 81

6 Genetic programming application on multi-objective optimisation 84
6.1 Motivation . . . 84
6.2 Methodology . . . 86
6.2.1 Selection and genetic operators for MOO2, αMOO2, and MOO3 . . . 86

6.2.2 α-dominance strategy αMOO2 . . . 87
6.2.3 Designating the final solution . . . 88
6.3 Experimental set-up . . . 90
6.3.1 Benchmarks . . . 90
6.3.1.1 MOO2 . . . 90
6.3.1.2 MOO3 . . . 90
6.4 Results and analysis . . . 90
6.4.1 MOO2 . . . 91
6.4.1.1 Pareto front . . . 91
6.4.1.2 Comparison of MOO2 to corresponding SOO approaches . . 91
6.4.1.3 Comparison of MOO2 algorithms to technical analysis trading strategies . . . 96
6.4.1.4 Buy-and-hold . . . 98
6.4.2 MOO3 . . . 100
6.4.2.1 Trader preference scenarios . . . 100
6.4.2.2 Pareto front . . . 101
6.4.2.3 Comparison of MOO3 to corresponding SOO approaches . . 103
6.4.2.4 Comparison of MOO3 algorithms to other technical indicators 112
6.4.2.5 Buy-and-hold . . . 115
6.4.3 Comparison of MOO2 algorithms to MOO3 algorithms . . . 115
6.5 Conclusion . . . 117

7 Thesis Contributions **122**

7.1 Summary of application on physical time scale . . . 122
7.1.1 Motivation . . . 122
7.1.2 Originality . . . 123
7.1.3 Findings . . . 123
7.2 Summary of application to event-based framework . . . 123
7.2.1 Motivation . . . 123

7.2.2 Originality 124
7.2.3 Findings 124
7.3 Summary of application of multi-objective optimisation 124
7.3.1 Motivation 124
7.3.2 Originality 125
7.3.3 Findings 125
7.4 Future work 125

8 Appendix A **128**
8.1 Physical time indicators 128

List of Figures

2.1 An example of DC. The grey line indicates the physical time series, the red line denotes a series of DC and OS events as defined by a threshold of 1%, while the green line denotes a series of DC and OS events as defined by a threshold of 1.5%. DC events are depicted with solid lines, while dotted lines denote the OS events. 12

2.2 Examples for two individuals having maximum depth 2 using the 'full' initialisation method. 20

2.3 Examples for two individuals having maximum depth 2 using the 'grow' initialisation method. 20

2.4 Examples for crossover between two individuals. 21

2.5 Example of Pareto fronts using $\mathbb{E}[RoR]$ and Risk as the objectives. 28

2.6 An example of crowding distance. 28

4.1 Example of the GP tree structure and the if-then-else structure 49

5.1 An example tree for GP-PT, GP-DC, and GP-DC-PT. 64

6.1 Comparison of the Pareto front solutions with the best single objective optimisation GP. 91

6.2 Representative run of the MOO3 and SOO algorithms for Apple stock. The SOO and MOO3-designated solutions are overlaid onto the MOO3 Pareto front for all seven scenarios considered. Risk, expected rate of return, and total return are shown here in their original units, rather than the normalised ones used during fitness evaluation. 102

List of Tables

4.1 Technical indicators and the corresponding periods used in this thesis. 48

4.2 Configuration of the GP algorithm 49

4.3 Selected parameters of the GP algorithm after parameter tuning 54

4.4 Average (standard deviation) $\mathbb{E}[RoR]$ and risk results of GP and other ML algorithms. Best value per metric is shown in boldface. 55

4.5 Kolmogorov-Smirnov tests between GP (control) and nine ML algorithms (first column) for $\mathbb{E}[RoR]$ and risk. p-values (second column) below the adjusted significance level (third column) appear in boldface to indicate statistical significance at 5% level. The calculation of the adjusted significance level is shown in brackets in the third column. 57

4.6 GP's average (standard deviation) performance under different stock markets. Best value per financial metric is shown in boldface. 57

4.7 GP's average performance under different countries 58

4.8 Average result for GP on 5 years versus 10 years 58

5.1 DC indicators; see also [5] and [6] 63

5.2 Physical time (technical analysis) indicators; see also [7] 64

5.3 Summary statistics of the GP-based algorithms on 5 and 10-year periods. We use boldface for the best values for each metric. 68

5.4 Sharpe ratio of the GP-based algorithms on 5 and 10-year periods. We use boldface for the best values for each period. 68

5.5 Kolmogorov-Smirnov tests between GP-DC (control) with GP-PT and GP-DC-PT for TR, $\mathbb{E}[RoR]$, and risk on 5 years. p-values (second column) below the adjusted significance level (third column) appear in boldface to indicate statistical significance at a 5% level. The calculation of the adjusted significance level is shown in brackets in the third column. 70

5.6 Kolmogorov-Smirnov tests between GP-DC (control) with GP-PT and GP-DC-PT for TR, $\mathbb{E}[RoR]$, and risk on 10 years. p-values (second column) below the adjusted significance level (third column) appear in boldface to indicate statistical significance at 5% level. The calculation of the adjusted significance level is shown in brackets in the third column. 70

5.7 Summary statistics of the GP-DC-PT algorithm and other nine machine learnings on a 5-year period. Best value per row is denoted in boldface. 73

5.8 Summary statistics of the GP-DC-PT algorithm and other nine machine learnings on a 10-year period. Best value per row is denoted in boldface. 74

5.9 Friedman with Bonferroni's post-hoc test between GP-DC-PT algorithms and ML algorithms on a 5-year period. Statistically significant differences at a 5% level are shown in boldface. 75

5.10 Total return 75

5.11 Expected Rate of Return 75

5.12 Risk 75

5.13 Friedman with Bonferroni's post-hoc test between GP-DC-PT algorithms and ML algorithms on a 10-year period. Statistically significant differences at a 5% level are shown in boldface. 76

5.14 Total return 76

5.15 Expected Rate of Return 76

5.16 Risk 76

5.17 Summary statistics of the GP-DC-PT algorithm and technical analysis on a 5-year period. Best value per row is denoted in boldface. 78

5.18 Summary statistics of the GP-DC-PT algorithms and technical indicators over a 10-year period. Best value per row is denoted in boldface. 79

5.19 Kolmogorov-Smirnov tests between GP-DC-PT (control) and three technical indicators for TR, $\mathbb{E}[RoR]$, and risk on a 5-year period. p-values (second column) below the adjusted significance level (third column) appear in boldface to indicate statistical significance at the 5% level. The calculation of the adjusted significance level is shown in brackets in the third column. 80

5.20 Kolmogorov-Smirnov tests between GP-DC-PT (control) and three technical indicators for TR, $\mathbb{E}[RoR]$, and risk on a 10-year period. p-values (second column) below the adjusted significance level (third column) appear in boldface to indicate statistical significance at the 5% level. The calculation of the adjusted significance level is shown in brackets in the third column. 81

5.21 Summary statistics between GP-DC-PT algorithm with buy-and-hold on the 5-year period. We use boldface for the best values for each measure. 81

5.22 Summary statistics between GP-DC-PT algorithm with buy-and-hold on the 10-year period. We use boldface for the best values for each measure. 82

5.23 Kolmogorov-Smirnov tests between GP-DC-PT (control) and buy-and-hold strategy for TR, $\mathbb{E}[RoR]$, and risk on a 10-year period. p-values below 0.05 appear in boldface to indicate statistical significance at the 5% level. 82

6.1 Comparison between SOO and MOO2 algorithms on 5-year periods. Best values per row appear in boldface. 94

6.2 Comparison between SOO and MOO2 algorithms on 10-year periods. Best values per row appear in boldface. 95

6.3 Kolmogorov-Smirnov tests between SOO (control) and MOO2 algorithms (first column) for $\mathbb{E}[RoR]$ and risk over a 5-year period. p-values (second column) below the adjusted significance level (third column) appear in boldface to indicate statistical significance at the 5% level. The calculation of the adjusted significance level is shown in brackets in the third column. 96

6.4 Kolmogorov-Smirnov tests between SOO (control) and MOO2 algorithms (first column) for $\mathbb{E}[RoR]$ and risk over a 10-year period. p-values (second column) below the adjusted significance level (third column) appear in boldface to indicate statistical significance at the 5% level. The calculation of the adjusted significance level is shown in brackets in the third column. 96

6.5 Summary statistics of the best two $\alpha MOO2$ algorithms and technical indicators over a 5-year period. Best value per row is denoted in boldface. 97

6.6 Summary statistics of the best two $\alpha MOO2$ algorithms and technical indicators. Best value per row is denoted in boldface. 98

6.7 Friedman with Bonferroni's post-hoc test between $\alpha MOO2$ and technical indicators. Statistically significant differences at 5% level shown in boldface. . . 99

6.8 Expected Rate of Return 99

6.9 Risk 99

6.10 Summary statistics for the best two MOO2 algorithms and buy-and-hold in terms of total return on the 5-year period. Best value per row appears in boldface. 99

6.11 Summary statistics for the best two MOO2 algorithms and buy-and-hold in terms of total return over the 10-year period. The best value per row appears in boldface. 99

6.12 Friedman test with Bonferroni's post-hoc test between $\alpha MOO2$ and buy-and-hold over the 5-year period. Statistically significant differences at the 5% level are shown in boldface. 100

6.13 Friedman test with Bonferroni's post-hoc test between $\alpha MOO2$ and buy-and-hold over the 10-year period. Statistically significant differences at the 5% level are shown in boldface. 100

6.14 Summary statistics between SOO and MOO3 for a 5-year period. The best value for each respective metric is shown in boldface. The values of a, b, c correspond to the weights as defined in Section 6.4.2.1. 105

6.15 Summary statistics between SOO and MOO3 in a 10-year period. The best value for each respective metric is shown in boldface. The values of a, b, c correspond to the weights as defined in Section 6.4.2.1. 106

6.16 Kolmogorov-Smirnov test results p-values for different weight set-ups ($[a, b, c]$) for a 5-year period. Statistically significant results at the 5% level are denoted in boldface. 107

6.17 Kolmogorov-Smirnov test results p-values for different weight set-ups ($[a, b, c]$) across a 10-year period. Statistically significant results at the 5% level are denoted in boldface. 108

6.18 Statistical test results for average TR, $\mathbb{E}[RoR]$, Risk, and Fitness function, according to the non-parametric Friedman test with the Hommel post-hoc test of different MOO and SOO algorithms across the 5-year period. The subscript for each algorithm denotes which metrics were optimised. When more than one metric is present, equal weights have been assigned to each metric. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface, indicating that the adjusted p-value is less than 0.05. 109

6.19 Statistical test results for average TR, $\mathbb{E}[RoR]$, Risk, and Fitness function, according to the non-parametric Friedman test with the Hommel post-hoc test of different MOO and SOO algorithms over the 10-year period. The subscript for each algorithm denotes which metrics were optimised. When more than one metric is present, equal weights have been assigned to each metric. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface, indicating that the adjusted p-value is lower than 0.05. 110

6.20 Summary statistics of the three-objective optimisation algorithms and technical indicators over the 5-year period. We use boldface for the best values for each measure. 113

6.21 Summary statistics of the three-objective optimisation algorithms and technical indicators over the 10-year period. We use boldface for the best values for each measure. 113

6.22 Statistical test results between MOO3 algorithms and technical analysis indicators for average TR, $\mathbb{E}[RoR]$, Risk, and Fitness function, according to the non-parametric Friedman test with the Hommel post-hoc test of different MOO and SOO algorithms over the 5-year period. The subscript for each algorithm denotes which metrics were optimised. When more than one metric is present, equal weights have been assigned to each metric. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface indicating that the adjusted p-value is lower than 0.05. 114

6.23 Statistical test results between MOO3 algorithms and technical analysis indicators for average TR, $\mathbb{E}[RoR]$, Risk, and Fitness function, according to the non-parametric Friedman test with the Hommel post-hoc test of different MOO and SOO algorithms over the 10-year period. The subscript for each algorithm denotes which metrics were optimised. When more than one metric is present, equal weights have been assigned to each metric. Significant differences arising between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface, indicating that the adjusted p-value is lower than 0.05. 114

6.24 Summary statistics for the three-objective optimisation algorithms and buy-and-hold in terms of total return over the 5-year period. The best values per metric appear in boldface. 115

6.25 Statistical test results between MOO3 algorithms and technical analysis indicators for average TR, $\mathbb{E}[RoR]$, Risk, and Fitness function, according to the non-parametric Friedman test with the Hommel post-hoc test of different MOO and SOO algorithms. The subscript for each algorithm denotes which metrics were optimised. When more than one metric is present, equal weights have been assigned to each metric. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface, indicating that the adjusted p-value is lower than 0.05. 116

6.26 Summary statistics for the three-objective optimisation algorithms and buy-and-hold in terms of total return over the 10-year period. The best values per metric appear in boldface. 116

6.27 Statistical test results between MOO3 algorithms and technical analysis indicators for average TR, $\mathbb{E}[RoR]$, Risk, and Fitness function, according to the non-parametric Friedman test with the Hommel post-hoc test of different MOO and SOO algorithms. The subscript for each algorithm denotes which metrics were optimised. When more than one metric is present, equal weights have been assigned to each metric. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface, indicating that the adjusted p-value is lower than 0.05. 116

6.28 Comparison between MOO2 and MOO3 algorithms on 5-year periods. Best values per row appear in boldface. 118

6.29 Comparison between MOO2 and MOO3 algorithms on 10-year periods. Best values per row appear in boldface. 119

6.30 Friedman with Bonferroni's post-hoc test between MOO3 algorithms and MOO2 algorithms across a 5-year period. Statistically significant differences at the 5% level are shown in boldface. 120

6.31 Total return 120

6.32 Expected Rate of Return 120

6.33 Risk 120

6.34 Friedman with Bonferroni's post-hoc test between MOO3 algorithms and MOO2 algorithms over the 10-year period. Statistically significant differences at the 5% level are shown in boldface. 121
6.35 Total return 121
6.36 Expected Rate of Return 121
6.37 Risk 121

Chapter 1

Introduction

Algorithmic trading refers to the process of executing financial transactions using algorithms that follow pre-defined rules. The applications of algorithmic trading involve various financial instruments, such as stocks, bonds, currencies, and derivatives. Due to its capability of analysing data and executing trades at speeds and frequencies that are impossible for human traders, algorithmic trading has become a focal point of investigation within the financial domain [8].

This rise in the popularity of algorithmic trading is attributed to several factors. First, traders seek high profits with low risk in financial trading. Second, the unpredictability and complexity of the market make it hard to predict future price movements. Third, according to the efficient market hypothesis (EMH), financial markets are highly competitive and all available information is therefore quickly reflected in asset prices. This rapid reflection makes any advantage gained by an existing strategy short-lived as more traders adopt it, forcing traders to continuously search for more advanced approaches.

An important component of algorithmic trading is financial forecasting. The main goal of financial forecasting is to provide insights and predictions about future market conditions, such as stock prices or market trends, helping traders make informed decisions. One traditional method used in financial forecasting is technical analysis. Technical analysis is a method that predicts the future price movement of securities by analysing historical market data, primarily price and volume. Unlike fundamental analysis which focuses on a com-

pany's financial health, the belief underlying technical analysis is that past trading activity and price changes can indicate future market behaviour. Technical analysis involves the use of various tools and techniques, including chart patterns, technical indicators, and statistical measures, to identify trends and potential reversal points. In addition, it can be incorporated and enhanced by ML techniques. Nowadays, many algorithmic trading systems use technical analysis, specifically utilising ML algorithms trained by technical indicators, to make trading decisions automatically. These technical indicators help automated trading systems to summarise useful information and filter out noise in the market. In this thesis, we investigate the application of ML using technical indicators in the field of financial forecasting.

1.1 Motivation

In the field of algorithmic trading, ML techniques are frequently employed alongside technical indicators, summarising information in terms of physical time series data to forecast future price movements. In particular, genetic programming (GP), an evolutionary technology that applies the Darwinian principle of evolution to improve its models, has been a popular ML algorithm with proven success in improving financial forecasting [9][10][11]. This combination leverages historical market data to identify patterns and trends, enhancing the accuracy of predictive models. The performance of such machine learning algorithms depends on many factors, including data analysis from different markets, data periods, forecasting days ahead, transaction costs, and benchmarks. For example, many references in the ML literature indicate that more data can allow an ML algorithm to better generalise; other works suggest that old data may be irrelevant in financial problems [12]. However, most of these factors have been neglected in previous studies. In addition, GP has been used successfully in financial forecasting in many cases and yet has only been compared with a few benchmark algorithms, usually less than four [13] [10], leading to reduced comparability and incomplete understanding of performance.

Therefore, the first contribution of this thesis is the investigation of the performance of a GP algorithm, benchmarked against nine ML algorithms, by considering the factors that

influence financial results, as detailed in Chapter 4.

The technical indicators used in the above application of the GP algorithm are calculated by the historical data under fixed time intervals, daily specifically, representing a physical time series prediction. As its name implies, the research in physical time series prediction relies on fixed physical time scales, such as daily, weekly, or monthly intervals. However, this conventional approach often leads to discontinuous data, resulting in the loss of significant information between data points. For example, if one uses daily closing prices, they would obtain only a single price point per day, ignoring all other price changes that occur during the day. All these ignored data points are often missed opportunities, both from a financial perspective (e.g., a trading algorithm could take advantage of these intraday price changes and make a significant profit) and also from an ML perspective since these additional data could have been used for training an algorithm. To avoid this issue, one could start using high-frequency data, such as hourly, or even at higher minute-by-minute intervals. However, the same problem occurs (albeit on a smaller scale) as summarising data under fixed time intervals ignores what happens in between. Alternatively, one could use tick-by-tick data, i.e., record every individual price change. While this offers the most accurate and detailed view of market activity, it introduces its own challenges, such as the large amounts of data and its intrinsic computational cost, as acquiring and maintaining tick-by-tick data can be expensive due to its high frequency and volume. Another disadvantage of tick data is that it can introduce 'noise' due to the extremely high number of data points.

To avoid the challenges that derive from the use of physical time, one could represent price movements as discrete events, effectively capturing noteworthy price fluctuations which are typically defined by thresholds, e.g., a 5% change. As a result, data is summarised by capturing significant market activities. The framework of directional changes (DC) introduced in [14] and formalised in [9], leverages a user-defined threshold value, denoted as θ, to detect substantial price movements. In the DC framework, a price series is partitioned into distinct upward and downward trends, marked by DC events when prices deviate beyond θ. These DC events are then followed by overshoot events, signifying price movements beyond the DC event. Therefore, directional changes tend to emphasise the magnitude of price changes

over varying time intervals, unlike the traditional approach under physical time which uses fixed intervals (e.g., daily closing prices). This concept offers traders a fresh perspective when analysing price movements, enabling them to concentrate on significant price changes while filtering out less relevant price details.

As a novel method, the application of financial forecasting under the DC framework remains unexplored. Furthermore, GP has demonstrated notable efficiency in combining diverse indicators for profitable trading strategies. As a result, we are interested in investigating the advantages and disadvantages of using GP under the DC framework, which constitutes the second contribution of this thesis and is presented in Chapter 5.

To balance the return against the relative risk in the trading strategy, we use the Sharpe ratio as the fitness function of the proposed GP-based algorithms. The Sharpe ratio can evaluate the risk-adjusted return, ensuring that the models optimise return and risk simultaneously. However, a drawback of the Sharpe ratio and similar aggregate fitness functions is that it may cause misunderstanding of the complex relationship arising among the different objectives and makes GP focus on optimising one part of the fraction only, e.g. only the numerator (return) or the denominator (risk). Such behaviour can still result in increased Sharpe ratio values as the evolutionary process continues, but without necessarily improving both return and risk.

To overcome this problem, we propose multi-objective optimisation (MOO) techniques that provide a framework for simultaneously optimising the expected rate of return and risk instead of using an aggregate fitness function. This approach yields a set of Pareto-optimal solutions that offer various trade-offs. Based on the same physical time and DC indicators, the proposed MOO algorithm adopts the well-known non-dominated sorting genetic algorithm II (NSGA-II) which follows the framework of the genetic algorithm (GA). Since both GA and GP belong to the evolutionary algorithm family and follow the same process, we use GP instead of GA in the NSGA-II algorithm. While NSGA-II has been very successful, its strict Pareto dominance criterion can lead to some solutions remaining in the later generations during NSGA-II, even when they exhibit extreme superiority in one objective but perform poorly in others. Therefore, we apply an α-dominance strategy that relaxes the strict dominance

criteria [15]. Furthermore, to evaluate the trading strategy in terms of long-term profitability, we extend our work by adding a third metric in the MOO framework, namely total return.

As a result, in the final contribution of this thesis, we propose two novel MOO approaches that use a GP-version of NSGA-II to optimise two and three fitness functions, namely MOO2 and MOO3. Both of these novel MOO algorithms proposed algorithms are trained using a combination of PT and DC indicators.

1.2 Thesis overview

The remainder of this thesis is structured as follows. Chapter 2 presents the background information to this thesis, including financial forecasting under physical time series and DC framework, GP, and MOO. Chapter 3 introduces the research related to physical time series predictions, GP, DC, and MOO approaches. Through the literature, we find the gap that motivates the experiments of this thesis. Then, Chapter 4 presents the first contribution of this thesis, investigating the in-depth comparison between genetic programming and 9 other ML algorithms. In Chapter 5, we explore the application of the GP under the DC framework. Chapter 6 extends the previous research into the field of multi-objective optimisation. Finally, Chapter 7 concludes this thesis.

Chapter 2

Background

In this chapter, we first introduce the background information of financial forecasting in Section 2.1, covering both physical time series and event-based predictions. Meanwhile, the technical indicators of physical time series and directional changes used in this thesis are introduced in Section 2.1.2 and Section 2.1.4 respectively. Second, Section 2.2 is dedicated to providing information on genetic programming, including representation, population initialisation, genetic operators, breeding methods, and selection. Third, we introduce the multi-objective optimisation method in Section 2.3, mainly presenting a well-known algorithm, NSGA-II, in Section 2.3.1. In addition, we also introduce a novel method that relaxes the dominance criteria of the NSGA-II in Section 2.3.2. Finally, Section 2.4 concludes this chapter.

2.1 Financial forecasting

Financial forecasting has always played a vital role in the financial world. To obtain the greatest trading returns with the least risk, financial traders hope to predict market trends and reversal points.

The goal of financial forecasting is to gain profit by making correct trading decisions. Usually, the decisions are made based on the opportunity caused by the information gap. According to the information relied on, financial forecasting can be broadly categorised into two primary methodologies, namely fundamental analysis and technical analysis. Fundamental

analysis focuses on economic indicators, financial statements, and various macroeconomic factors to assess the intrinsic value of securities. On the other hand, technical analysis, which is one of the techniques used in this thesis, relies on the study of past market data, primarily price and volume, to forecast future market behaviour.

The traditional method of conducting technical analysis relies on fixed interval data series, also known as physical time series prediction. Moreover, over the past decade, an alternative approach has emerged, one that focuses on key events and which has garnered significant attention and success in the literature. This event-based approach, specifically directional changes (DC), as discussed in this thesis, transforms historical time series data into a sequence of events. Further details on these methodologies are presented below.

2.1.1 Physical time series prediction

Most financial forecasting relies on the physical time scale, such as hourly, daily, or weekly data. This approach relies on the principle that future price movements could be reflected in the historical data. Given its sole reliance on historical data for prediction, technical analysis is widely used in physical time series prediction.

Technical analysis, as one of the most traditional methods of evaluating stocks, is used by approximately 90% of the traders [16]. The main idea of technical analysis is the use of charts and graphs integrated with various statistical methods to predict the market trend and stock price [17]. In textbooks of technical analysis, there are three rules to guide traders [18]. The first one is that the action of the market price reflects all the information. In other words, the price movement is derived from all relevant financial information. The second rule is about the trend of price movement. The final goal of technical analysts is to find the trend and the time at which the trend starts to reverse. It enables traders to make a profit from selling stocks in the downtrend and buying stocks in the uptrend. The last rule is that history repeats itself. A big assumption behind technical analysis is that the same event happens under the same conditions. Thus, traders tend to make the same decisions when all conditions are similar. However, there has been some doubt whether technical analysis can yield unusual profits based on past prices. This is likely to be against the EMH, which asserts

that share prices reflect all available information and that stocks always trade at their fair value. Consequently, it is impossible to purchase undervalued stocks or sell stocks at inflated prices. In other words, no one can achieve excess profit without taking on corresponding risks. Therefore, more and more studies have been developed to prove whether or not technical analysis is profitable. As an example, a study in [19] found no evidence to confirm that technical analysis can earn excess profit in the stock market at the very early stages. However, with further research development, newer and more studies provided strong evidence for the profitability of technical analysis [20] [21].

There are two main types of technical analysis. The first charts patterns - a subjective form of technical analysis which allows traders to look at the specific period past the target price and figure out patterns based on skill and experience [18]. The second arises in the form of technical indicators that provide clear and rule-based signals for trading. By mathematical calculation, the original data is converted into a value that measures the data's different characteristics. Based on the indicators, traders can ignore the noise in the data and make decisions as to when to buy and when to sell. As an example, the most common indicator used by traders is the moving average, which smooths the historical price to help traders spot trends more easily. In real trading, various indicators are adopted simultaneously to reduce noise.

2.1.2 Physical time series indicators

There are several physical time technical indicators that traders can use in the stock market. Below, we introduce the main indicators which are used in this thesis. These indicators are widely used in the literature [7] [22] [23]. Since this indicator is not the primary focus of the thesis, the remaining indicators are introduced in Appendix A.

1) *Relative Strength Index (RSI)*

   RSI is a momentum oscillator that measures the magnitude of recent price changes and determines an asset's overbought or oversold conditions. It is commonly used with another indicator, MACD, introduced in Equation A58, to make predictions. The formula

is presented in Equation 2.1.

$$RSI = 100 - (\frac{100}{(1 + \frac{AvgU}{AvgD})}) \tag{2.1}$$

where AvgU/AvgD is the average percentage gain/loss over a selected period. RSI will return a number between 0 and 100. Traditionally, a trade is considered overbought when it is above 70 and oversold when it is below 30.

2) *Moving average (MA)*

MA is the moving average for a given period. The formula is presented in Equation 2.2.

$$MA = \frac{\sum_{i=1}^{n} A_i}{n} \tag{2.2}$$

where A_n stands for the price of the day and n is the period of the MA specified by the trader.

3) *Commodity Channel Index (CCI)*

CCI measures the deviation of an asset's price from its statistical average. The formulas are presented in Equation 2.3 to 2.5.

$$CCI = \frac{TP - MATP}{0.015 \times MeanDeviation} \tag{2.3}$$

$$TP = \frac{H + L + C}{3} \tag{2.4}$$

$$MeanDeviation = \frac{\sum_{i=1}^{n} \mid TP - MATP \mid}{n} \tag{2.5}$$

where TP stands for the typical price. The H, L, and C are the high price, low price, and close price, respectively. The high price is the highest stock price during a trade session and the low price is the lowest stock price during a trade session. The close price is the price at the end of the last trading session, while MATP is the moving average of the TP.

4) *William's %R*

William's %R measures oversold or overbought conditions of an asset by comparing the closing price of an asset to its price range over a set period. The formula is presented in Equation 2.6.

$$William's\ \%R = -100 \times \frac{HighofRange - C}{HighofRange - LowofRange} \tag{2.6}$$

where HighofRange and LowofRange are the highest price and lowest price on a specified period and C is the close price.

5) *Average True Range (ATR)*

ATR is a volatility indicator developed by [24] and measures the volatility of an asset by calculating the average of the true range over a set period. The formulas are presented in Equations 2.7 and 2.8.

$$ATR = \frac{\sum_{i=1}^{n} TR_i}{n} \tag{2.7}$$

$$TR = MAX[(H - L), (\mid H - C \mid), (\mid L - C \mid)] \tag{2.8}$$

where H, L, and C stand for the high, low, and close prices, respectively.

6) *Exponential Moving Average (EMA)*

EMA calculates a weighted average of a series of prices over a set period, where more recent prices are given greater weight in the calculation. The formula is presented in Equation 2.9.

$$EMA_i = C \times \frac{2}{1+n} + EMA_{i-1} \times (1 - \frac{2}{1+n}) \tag{2.9}$$

where EMA_{i-1} is the previous EMA and n represents the look-back period.

7) *On Balance Volume (OBV)*

OBV measures buying and selling pressure and calculates the cumulative total of an asset's volume, where positive volume is added to the total of an 'up' day and negative volume is subtracted on a 'down' day. The formula is presented in Equation 2.10.

$$OBV_i = \begin{cases} OBV_(i-1) + volume, & \text{if } C > C_{i-1} \\ OBV_(i-1) - volume, & \text{if } C < C_{i-1} \end{cases} \tag{2.10}$$

where OBV_{i-1} and C_{i-1} are the previous OBV and C, respectively.

8) *Parabolic Stop and Reverse (PSAR)*

PSAR identifies potential reversals in the direction of an asset's price movement by placing dots on a chart that indicate potential stop and reverse points for a long or short position. The formula is presented in Equation 2.11.

$$PSAR_i = PSAR_{i-1} + AF(EP - PSAR_{i-1}) \tag{2.11}$$

where AF is an acceleration factor that starts at 0 and increases by 0.01 per data point. EP, which stands for the extreme point, is the lowest low price in the current downtrend and the highest high price in the current uptrend. When PSAR is below the current high price, the trend is an uptrend; otherwise, it is a downtrend. When the trend switches, the value of AF is reset to 0. The initial PSAR is set to the first low price.

2.1.3 Event-based prediction

In 1967, [25] proposed the idea that the physical time scale is not necessarily the fundamental scale for the analysis and forecasting of the financial market. Instead, an event-based approach, focusing on the important event, was proposed as an innovative way of summarising price movements. Event-based prediction, as its name implies, summarises the time series data into a series of discrete events. There are many types of event-based methods, including zigzag[26] and turning point [27].

In this thesis, we focus on a specific subset of event-based methods known as directional changes (DCs). DCs form an event-based approach for summarising market price movements, as opposed to a fixed-interval-based approach. A DC event is identified only when the price movement of the time series exceeds a threshold as pre-defined by the trader. Depending on

the direction of price movement, such DC events could be either upturn events or downturn events. Frequently, after the confirmation of a DC event, an overshoot (OS) event follows and the OS event ends when a new price movement starts in an opposite trend, eventually leading to a new DC event.

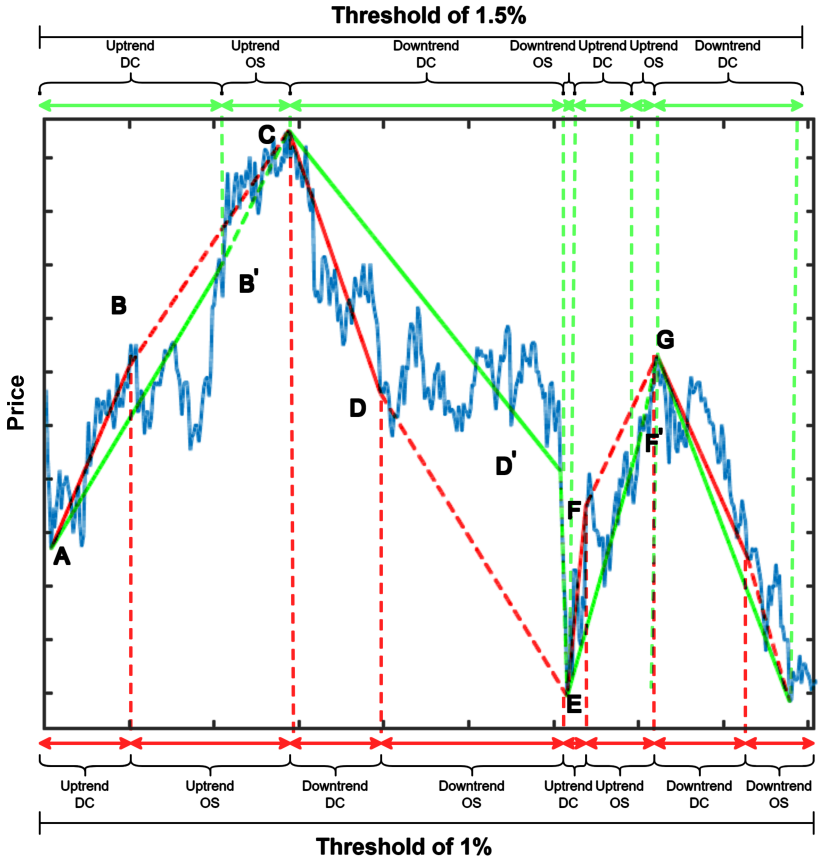


Figure 2.1: An example of DC. The grey line indicates the physical time series, the red line denotes a series of DC and OS events as defined by a threshold of 1%, while the green line denotes a series of DC and OS events as defined by a threshold of 1.5%. DC events are depicted with solid lines, while dotted lines denote the OS events.

Figure 2.1 presents an example of how to convert physical time series to DC and OS events using two different thresholds (see red and blue lines). Note that thresholds may, in principle, vary, as traders need not necessarily agree on which price movement constitutes a significant event whereupon each such threshold leads to a different event series. A smaller threshold leads to the identification of more events and increases the opportunity for trade, while a larger threshold leads to fewer events with greater price movement. Thus, selecting

an appropriate threshold is a key challenge.

By looking at the historical daily price movement (grey line) and the events created by the threshold of 1% (red lines), there are plenty of price movements that are not classified as events under the DC framework, as these do not exceed the threshold. Only when a price change is larger than the threshold is the time series divided into DC events (solid lines) and OS events (dotted lines). For example, the solid red line from A to B is considered a DC event on an upturn, while an OS event follows (from B to C) and then a new DC event (in the opposite direction) is detected from C to D, and this is followed by an OS event from D to E in a downturn, and so on.

It is worth noting that the trend change can be confirmed only when the price movement exceeds the threshold. In other words, we will know when the OS event ends when the next DC event (in the opposite direction) is confirmed. For example, in Figure 2.1, point D is a DC event confirmation point. Before point D, the last OS event is considered to still be active, while the trader considers it to have been included in an upward event. This leads to a paradox that, on the one hand, in order to maximise returns, trades should be closed as near as possible to the endpoint of the OS event while, on the other hand, when the endpoint of the OS event is detected, it is already well beyond that point. Therefore, figuring out the extreme point where the direction is reversed, such as point C in Figure 2.1, remains an active research topic in the DC domain and, in particular, several scaling laws have been suggested to establish OS event length [28].

The advantage of DC is that it offers traders a new perspective on price movements as it allows them to focus on significant events and ignore other price movements that could be considered noise. Therefore, DC leads to new research directions and challenges that are not relevant under physical periods.

2.1.4 Directional changes indicators

Similar to having physical time series indicators, the DC paradigm includes its own indicators. This is an important benefit of DC, as its indicators are able to capture patterns that physical time series indicators cannot. Next, we present the DC indicators used in this thesis.

1) *Total price movements value at extreme points (TMV)*

TMV is defined as a price change between the extreme points defining the beginning and end of a trend as normalised by the threshold θ. The formula is presented in Equation 2.12.

$$TMV = \frac{EP_i - EP_{i-1}}{EP_{i-1} \times \theta} \tag{2.12}$$

where EP represents the extreme price of the current DC trend, EP_{i-1} represents the extreme price of the last DC trend, and θ is the user-specific threshold of the DC.

2) *Overshoot Values at Extreme Points (OSV)*

The OSV is the difference between the current price and the last DC confirmation point divided by the threshold θ. The formula is presented in Equation 2.13.

$$OSV = (P_c - P_{DCC}) \div P_{DCC} \div \theta \tag{2.13}$$

where P_c represents the current price, P_{DCC} is the last DC confirmation price, and θ is the threshold.

3) *Average OSV*

The average OSV is the average value of the OSV indicator over a given period. It is calculated as the Equation 2.14.

$$Average\ OSV = \frac{\sum_{i=1}^{n} OSV_i}{n} \tag{2.14}$$

4) *Time-adjusted return of DC (R_{DC})*

R_{DC} is equal to the product of TMV and the threshold θ, divided by the time intervals between each extreme point. The formula is presented in Equation 2.15.

$$R_{DC} = \frac{\mid TMV \mid \times \theta}{T} \tag{2.15}$$

where TMV states the total price movement value at extreme points, T is the time

interval between each EXT, and θ is the threshold used.

5) *Average time-adjusted return of DC (average R_{DC})*

Average R_{DC} is the average value of the R_{DC} indicator over the given period. The formula is presented in Equation 2.16.

$$Average\ R_{DC} = \frac{\sum_{i=1}^{n} R_{DC}}{n} \tag{2.16}$$

6) *Time for completion of a trend (T_{DC})*

T_{DC} is defined as the Duration of a DC trend and is calculated by the period spent through a DC trend.

7) *Average time for completion of a trend (average T_{DC})*

Average T_{DC} is the average value of the T_{DC} indicator over the given period. The formula is presented in Equation 2.17.

$$Average\ T_{DC} = \frac{\sum_{i=1}^{n} T_{DC}}{n} \tag{2.17}$$

8) *Number of directional change events (N_{DC})*

N_{DC} is defined as the total number of DC events over the given period.

9) *Time-independent Coastline (C_{DC})* C_{DC} is defined as the length of the price-curve coastline under DC, calculated by the sum of the absolute value of the TMV indicator over the given period. The formula is presented in Equation 2.18.

$$C_{DC} = \sum_{i=1}^{n} \mid TMV_i \mid \tag{2.18}$$

where n stands for the total number of DC events over the selected period.

10) *Up and down trend asymmetry in time intervals (A_T)*

Under the DC framework, the stock price is divided into a series of uptrends and downtrends, which follow each other. Unlike physical time series, the number of uptrends and downtrends is relatively equal, but their duration may vary. A_T represents the disparity between the time DC classifies as up trends and down trends over the given period. The formula is presented in Equation 2.19.

$$A_T = \frac{T_{m\uparrow} - T_{m\downarrow}}{T_{m\uparrow} + T_{m\downarrow}} \tag{2.19}$$

where $T_{m\uparrow}$ and $T_{m\downarrow}$ represent the median time spent on the uptrend and downtrend, respectively, over the given period.

More details of the DC indicators can be found on [5] and [6].

2.2 Genetic programming

2.2.1 General information

GP was first developed by [29] and has been used on financial forecasting problems for over 20 years [8]. It is a bio-inspired technique that evolves computer programs to tackle problems or execute tasks. GP incorporates key elements for an effective global search in that, instead of focusing on a single solution, GP operates with a population of candidate solutions (individuals). Moreover, the fitness function evaluates the quality of each individual in the population, favouring higher-quality solutions for progression to the next generation, while genetic operators explore the solution space by generating new offspring individuals through a stochastic selection process based on fitness.

As Algorithm 1 outlines, the GP process starts with a randomly generated population, which consists of p individuals, also known as population initialisation. These individuals are constructed using terminal sets, which include variables and constants, in addition to function sets that define operations for manipulating these elements, potentially encompassing arithmetic operations (+, -, $\times$, /), mathematical functions (sin, cos, exp, log), Boolean operations (AND, OR, NOT), conditional operators (If-Then-Else), iterative functions (Do-Until),

and comparison operators ($>$, $<$, $=$).

Following the establishment of the initial population, the algorithm evaluates a user-defined fitness function for each individual, quantifying their effectiveness. Subsequently, individuals are chosen to produce new offspring whereupon the individual with the higher fitness function has a greater chance of being selected, as per natural selection. The selected individuals are modified by genetic operators such as crossover and mutation, thereby introducing modifications for exploring the search space. Through the genetic operators, new offspring are generated and a new population comprising new offspring is seeded. In the new population, the fitness function is again calculated, and the selection process continues. The entire process, which includes fitness function calculation, individual selection, offspring breeding, and the creation of a new population, is termed a generation. In a single run of GP, this cycle of generations is repeated until a termination criterion is reached (a certain number of generations or an optimal fitness function value). This evolutionary approach allows GP to conduct a robust global search in the candidate solution space, thereby minimising the risk of being confined to local minima. Ultimately, the individual with the best fitness function is considered the result of the GP. It is worth noting that the population initialisation is only processed once at the outset of the GP.

2.2.2 Genetic programming representations

As mentioned before, the theory of GP comes from the process of natural evolution. GP representation refers to the method used to symbolise the genotype[1] of individuals. This representation is crucial not only for facilitating the evolutionary process but also for ensuring that the resulting solutions are interpretable.

Back in the early stages of GP development, the representation of individuals was based on LISP S-expressions. This approach, rooted in the LISP programming language, employs a uniform list format to represent both data and code, effectively capturing the essence of individuals within the GP framework. Later on, [29] refined this concept by abstracting LISP S-expressions into a tree-based representation that transcends the limitations of any single

[1] Genotype is the strategy of the solution that GP aims to solve.

Algorithm 1 High-level pseudocode of a genetic programming algorithm.

GP(p, *Fitness*, *pc*, *pm*)
  p: population size
  Fitness: determines the quality of solutions
  pc: crossover rate
1: *Initialise population*: $P \leftarrow$ *Generate p individuals (candidate solutions) at random*
2: *Evaluate*: **for each** i **in** P, calculate *Fitness(i)*
3: **while** termination condition not satisfied **do**
4: $P_g \leftarrow$ *Create new (empty) population for generation g*
5: *Mating*: probabilistically select p individuals from P. For each of these:
  i. Perform crossover between the best performer and second best performer if the probability condition is met
  ii. Perform mutation for the best performer if the probability condition is not met
  iii. Add the resulting offspring to P_g
6: *Evaluate*: **for each** i **in** P_g, calculate *Fitness(i)*
7: *Survival*: $P \leftarrow P_g$
8: **end while**
9: **Return** the individual with the highest fitness from P

programming language. This tree-based model, comprising both function sets and terminal sets as previously outlined, has become the predominant form of representation in contemporary GP applications, and it is this model that we adopt in our thesis. In addition, several different representations of the GP may be found, as follows: Linear Genetic Programming [30], Cartesian Genetic Programming [31], and Graph Genetic Programming[32].

2.2.3 Population Initialisation

The initialisation of the population is a critical step in the GP process, setting the stage for the performance of the evolutionary algorithm from the beginning. Despite occurring only once, the significance of this phase remains undiminished, as it fundamentally influences both the diversity and potential of the initial population. Within GP, there are three primary approaches to population initialisation: full, grow, and a combination of these two known as 'ramped half-and-half'.

To comprehend these methods fully, it is essential to introduce the concept of depth within the GP context. Depth refers to the number of layers in an individual's structure, extending from the root (considered depth 0) to the deepest terminal node. Therefore, the depth con-

trols the size and shape of the individual.

In both ‘full’ and ‘grow’ methods, the individuals are generated randomly without exceeding the maximum depth selected by the users. The ‘full’ method, as its name implies, allows an individual to generate a full tree, which in turn forces the depth of the node on each branch to equal the maximum depth. Therefore, in the ‘full’ method, each individual has the same depth and a symmetrical shape. However, this does not mean the individual has the same number of nodes, as it is also referred to as the size of the individual. It will happen only when all function sets have the same arity. Figure 2.2 shows two examples of individuals having maximum depth 2 using the ‘full’ initialisation method. We can observe that the right individual has one more node than the left because it contains a sin function, which has one arity, instead of the ‘+’ symbol, which has two arities.

Conversely, the ‘grow’ method introduces greater variability in the size and shape of individuals. Here, nodes are randomly selected from both function and terminal sets. Once a terminal set is selected, the branch ends. When the depth of the nodes reaches the maximum depth, the terminal set is forced to be selected, and the branch ends. The ‘grow’ method allows the individual to develop a branch with a lower depth. The individual could have a branch not reach the maximum depth, as Figure 2.3 shows. The individuals derived from the ‘grow’ method are controlled not only by the maximum depth but also by the proportion between the terminal sets and function sets. For example, when terminal sets predominate, the individuals can have very small sizes and shapes regardless of the maximum depth, because the likelihood of selecting terminal sets is very high. On the contrary, in the situation that the function sets are significantly greater than terminal sets, there will not be too much difference between the ‘grow’ and ‘full’ methods.

In addition, the combination of the ‘grow’ and ‘full’ methods was raised by [29]. It uses a range of depth limits instead of one maximum depth. Meanwhile, the ‘ramped half-and-half’ method forces half of the population to generate individuals following the ‘grow’ method and the other half to follow the ‘full’ method. This dual approach enhances the diversity of the initial population by incorporating individuals of varied sizes and shapes, thereby enriching the genetic pool available for the evolutionary process.

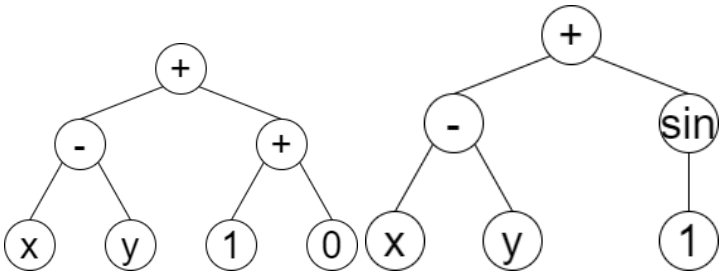


Figure 2.2: Examples for two individuals having maximum depth 2 using the 'full' initialisation method.

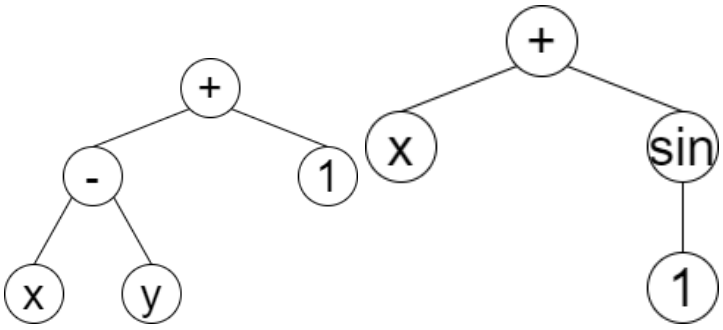


Figure 2.3: Examples for two individuals having maximum depth 2 using the 'grow' initialisation method.

2.2.4 Genetic Operators

Genetic operators play a pivotal role in guiding the evolution of individuals within the GP framework, ensuring both the generation of offspring and the maintenance of population diversity. The primary genetic operators include crossover, mutation, and reproduction, each serving distinct functions in the evolutionary process.

Crossover draws inspiration from biological reproduction, wherein offspring inherit genetic material from both parents. This genetic exchange aims to create new gene combinations that could enhance the offspring's ability to solve problems more effectively. Various crossover techniques exist, such as one-point, two-point, and uniform crossover. For illustrative purposes, we consider the one-point crossover, where two parents are selected, and a single crossover point is identified for each. Sub-trees from these crossover points are then swapped between the parents, resulting in the creation of two new offspring, as shown in Figure 2.4. While the two-point crossover involves two such points for exchange, the uniform crossover employs a probabilistic approach for node exchange, potentially increasing diversity

but also risking the disruption of parental strategies [33]. In this work, the one-point crossover is employed due to its simplicity and effectiveness.

Mutation, unlike crossover, introduces new genetic variations without altering the existing gene pool. It requires only a single parent to produce offspring, applying random modifications to the individual. Common mutation strategies include one-point mutation, whereby a single node is replaced, and sub-tree mutation, which replaces an entire sub-tree from a randomly chosen node. These mutations enable the preservation and enhancement of population diversity across generations.

Reproduction is the simplest genetic operator, involving the direct copying of an individual to the subsequent generation. This method is particularly useful for retaining high-performing individuals within the population. In our experiments, we implement elitism, a form of reproduction that ensures the individual with the highest fitness is carried over to the next generation, thereby guaranteeing the maintenance or improvement of the GP's performance over time.

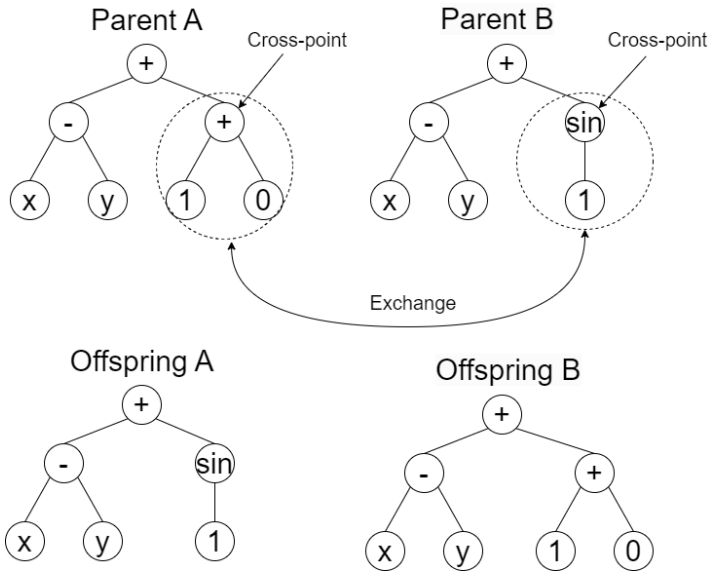


Figure 2.4: Examples for crossover between two individuals.

2.2.5 Breeding method

The breeding method in GP encompasses three critical components that decide the evolutionary progression of the population, specifically: 1) the strategy for constructing the subsequent population; 2) the restriction for mating selection; and, 3) the selection and application of genetic operators.

First we consider the strategy for constructing the subsequent population. The main replacement techniques include 'Generational' and 'Steady-State'. In the generational method, an empty population is created, and the offspring produced by either crossover or mutation are saved in the new population. This process of employing genetic operators continues until the new population matches the size of the existing population. Subsequently, the GP advances to the next generation, and the preceding population is discarded. Thus, in the generational method, the newly established population is comprised solely of the offspring. Conversely, the 'Steady-State' method diverges from the generational approach by maintaining continuity within the same population. The Steady-State indicates there is no generation and new population. All the selection and breeding processes occur within the original population. Specifically, offspring are integrated directly into the existing population, effectively replacing the existing individuals. This method ensures a constant population size through the immediate substitution of selected individuals with their offspring.

Additionally, there is another way to maintain the diversity of the population, which is restricting the mating target. Some examples are introduced in the following. The panmictic approach, as proposed by [34], permits unrestricted mating across the population. [35] introduced a concept of segregating populations into sub-groups, restricting mating to within these sub-populations. Moreover, [36] proposed the so-called 'age-layered population structure', which organises individuals into groups based on their age (the number of generations passed) and forces them to mate the individual in the neighbouring group (the last or next generations).

The final strategy pertains to the choice of genetic operators. [29] highlighted the use of crossover, which was later augmented by the incorporation of mutation, an approach that is increasingly favoured in research. The occurrence of crossover or mutation is determined by

a predefined probability. Traditionally, crossover, as the primary process, is assigned a higher probability (e.g., 95% in this thesis), leaving mutation to the remaining 5%. It is essential to note the operational difference wherein crossover necessitates two parents, whereas mutation requires only one, introducing a distinctive dynamic to the GP process.

2.2.6 Selection

The selection mechanism plays a crucial role in GP by dictating which individuals from the population are chosen for mating. This process is grounded in the principles of Darwinian evolution, wherein individuals exhibiting superior performance are afforded a greater chance of survival. That is achieved by the selection method. The majority of selection methods use the fitness function as the determining metric, including widely used methods such as tournament selection, which will be used in the experiments of this thesis, as well as the roulette wheel approach. The tournament method randomly selects a user-defined number of individuals. The individual with the best fitness function then becomes the parent for subsequent genetic operations. Conversely, the roulette wheel takes the whole population to be a 'wheel' in which each individual takes a certain part of the space according to the proportion of its fitness function value in the population. The process casts a 'dart' at this wheel, with those individuals occupying larger segments—indicative of higher fitness—having an increased probability of selection.

Beyond traditional fitness-based selection, alternative metrics can be employed to accommodate specific evolutionary objectives. An illustrative example is the NSGA-II, a multi-objective optimisation GA-based algorithm, one which utilises crowding distance as a selection criterion instead of fitness values. It calculates the distance of the individual relative to two neighbouring individuals within the solution space. NSGA-II is introduced in Section 2.3.1.

2.3 Multi-objective optimisation

In the real world, many problems require trade-off solutions among multiple conflicting objectives rather than focusing on a single objective. For instance, in the field of finance, traders seek a balance between profit and risk. They look for a trading strategy that combines high profit with low risk. In other words, they are solving a problem that maximises a profit objective and minimises a risk objective. It is vital to recognise that profit and risk represent two inherently conflicting objectives, where, traditionally, high profit is often associated with high risk, while low risk tends to correlate with reduced profit. Usually, traders consider profit and risk together, trying to achieve a good balance between the two. One way of tackling this problem is by using aggregate metrics. These involve combining all the different objectives into a single mathematical expression that is to be optimised directly as a single objective. A common example of such an aggregate metric is the Sharpe ratio [37], defined as the ratio of the expected rate of return over the risk. Fusing multiple objectives in this manner can be appealing, as this can simplify the evaluation of the genetic algorithm considerably. It also easily allows one to specify 'weights' for each objective, denoting the extent to which an investor values each of the different components of the problem. However, using such a predefined 'recipe' for condensing multiple factors into a single number, risks oversimplifying the complex relationship which underlies the different objectives, and thus increases the chances of misrepresenting the performance evaluation of the investment portfolios in question during the algorithmic process. On the other hand, in a multi-objective optimisation (MOO) approach, the different objectives (e.g., both return and risk) are optimised independently.

2.3.1 NSGA-II algorithm

NSGA-II [38] is a state-of-the-art genetic algorithm, effectively an extension of the traditional single-objective genetic algorithm approach, which allows the user to efficiently tackle problems with multiple, potentially conflicting objectives. NSGA-II is built on the GA framework. As an evolutionary algorithm inspired by the process of natural selection, GA follows almost the same process as GP. The main difference between GA and GP is in their representation,

wherein GA uses fixed-length strings, whereas GP uses tree structures. The tree structure in GP allows for the evolution of both the structure and parameters of the solutions, while in GA, the solution structure is fixed and only the values within the strings are evolved. In our experiments, we use NSGA-II which follows the GP process rather than GA.

Based on the original GP, NSGA-II employs three changes, namely (1) fast non-dominated sorting approach; (2) the crowding distance sorting; and, (3) a competition between parents and offspring. The pseudocode of NSGA-II is detailed in Algorithm 2. The primary process of this algorithm is introduced below.

2.3.1.1 Fast non-dominated sorting approach

To enforce a ranking among the pool of potential candidate solutions that comprise the Pareto fronts, NSGA-II implements a non-dominated sorting technique that relies on the concept of Pareto dominance. A candidate solution is said to Pareto-dominate another solution if it is better in at least one objective, and not worse in any other objective. This methodology entails sequentially building and removing each Pareto front from the initial population until no individuals remain. The detailed process is introduced below.

First, as with other evolutionary algorithms, NSGA-II generates a pre-defined quantity of individual solutions, represented as tree-based strategy, to form the original population. Following this, it computes and records multiple fitness functions for each individual, designated as f_1, f_2, f_3, and f_m, respectively, referring to m fitness functions that need to be optimised. Then, NSGA-II compares each individual and calculates the domination relationship, which is the number of the individual i that is dominated by others, marked as n_i. When n_i=0, it indicates that there are no individuals dominating the individual i. Therefore, these individuals with 0 n_i value are regarded as the first Pareto front, marked as r=1. They are deleted from the population and saved into the new population. Subsequent iterations replicate this process for the residual population, assembling successive Pareto fronts (r=2, and so forth) which are then integrated into the newly formed population. This iterative procedure continues until the initial population is empty, thereby dividing it into various Pareto fronts.

Through this methodical approach, NSGA-II efficiently divides the population into mul-

Algorithm 2 Pseudocode for NSGA-II

Require: population P, Pareto front rank r, individuals p and q, cross over probability k, a random number m (0<m<1).

Initialisation: Create the initial parent population by filling it with n randomly generated individuals
Evaluation: Evaluate all objectives for each of the individuals in the initial parent population
Fast Non-dominated Sorting:
while $P = \emptyset$ **do**
    for i $\in$ P **do**
        $r = r + 1$
        for q $\in$ P **do**
            if q dominates i **then**
                $n_i = n_i + 1$
            end if
        end for
        if $n_i = 0$ **then**
            $Pareto_r$ append i
            remove i from P
        end if
    end for
end while
Crowding Distance Calculation: Calculate the crowding distance for individuals of each Pareto front
while stopping criterion is not satisfied **do**
    Offspring population generation:
    while number of offspring population $\neq$ number of P **do**
        Parent selection: Apply tournament selection to select two individuals based on the r and crowding distance.
        if $m > k$ **then**
            Crossover: Apply the crossover operator to two individuals to produce two offspring
        else
            Mutation: Apply the mutation to two individuals to produce two offspring
        end if
    end while
    Evaluation: Evaluate all objectives for each of the individuals in both parent population and offspring population
    Fast Non-dominated Sorting: Apply Fast Non-dominated Sorting for both parent and offspring populations
    Crowding Distance Calculation: Calculate the crowding distance for individuals of each Pareto front
    A competition between parents and offspring: Rank all the individuals in both the parent population and offspring population and remove the half individuals with lower rank to keep same number of population
end while

tiple Pareto fronts, building a hierarchical structure of solutions based on their degree of non-domination. This approach helps identify the Pareto front of best solutions across different objectives, allowing for a detailed exploration of the solution space in multi-objective optimisation.

2.3.1.2 Crowding distance

Based on the concept of the Pareto front, it is clear that the individual in the front with a higher rank dominates the individuals in the front with a lower rank, while the individuals within the same Pareto front are regarded as being equivalent. Hence, within each front, no individual is dominated by other individuals. NSGA-II further ranks individuals within the same Pareto front by introducing the concept of crowding distance, which encourages the spread of solutions by considering the density of solutions around each candidate. The crowding distance is calculated as the normalised Manhattan distance between the two individuals closest to the solution of interest within the same front. This is effectively calculated as the sum of absolute differences between the two individuals, across the various objectives. An example of crowding distance calculation for a return-risk two-objective problem is shown in Figure 2.6, where points $i-1$ and $i+1$ are the two neighbouring points of point i within the same front. The generic formula for a crowding distance with k objectives is presented by Equation 2.20. It is worth noting that the individuals with the best value on a particular function are assigned a maximum crowding distance since they lack nearby individuals with comparable fitness values. First, NSGA-II divides the population into several Pareto fronts. For each front, NSGA-II calculates the crowding distance for each individual, assuming the maximum value for the individual has the best value on a particular objective. The higher crowding distance indicates better performance. Therefore, NSGA-II prioritises the Pareto front rank first and the value of crowding distance second, as Section 2.3.1.3 shows. In this way, NSGA-II effectively ranks each individual, aiding in the identification of promising solutions within the population, as shown in Figure 2.5.

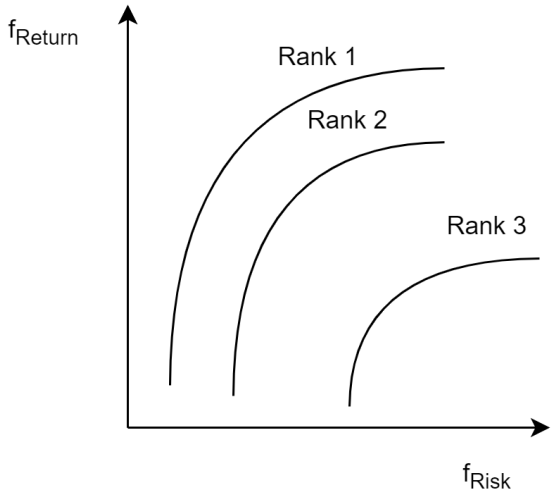


Figure 2.5: Example of Pareto fronts using $\mathbb{E}[RoR]$ and Risk as the objectives.

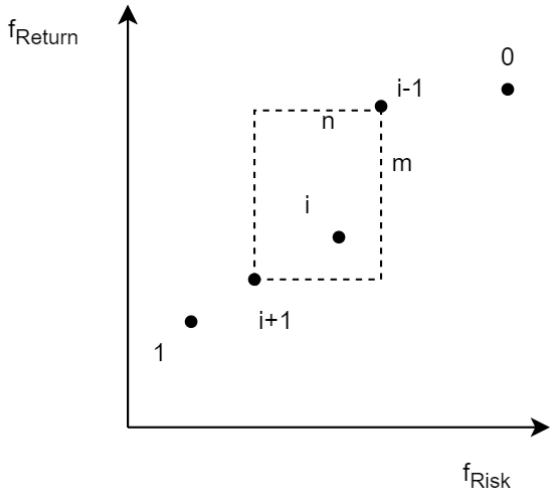


Figure 2.6: An example of crowding distance.

$$\text{Crowding distance for individual } i = \sum_{x=1}^{k} \left| \frac{f_x(i+1) \; - \; f_x(i-1)}{\max_j \left[f_x(j)\right] - \min_j \left[f_x(j)\right]} \right| \tag{2.20}$$

where f_x(i) represents how well the *i*-th individual in the population performs with respect to a specific objective 'x'. In addition, f_x(0) shows the performance of the individual when 'x' has its most favourable value, and f_x(1) corresponds to the individual with the least favourable 'x' value.

2.3.1.3 Competition of parents and offspring

During selection, NSGA-II first compares the Pareto front rank whereupon the individual with a lower rank survives. For the individuals with equal Pareto front rank, the one with a higher crowding distance is selected.

After a new population has been formed through genetic operators, both the original/parent and new population are evaluated and ranked, enabling the selection of the better half of individuals based on their Pareto front rank and crowding distance. Eventually, an equal number of individuals with lower Pareto front rank and higher crowding distance survive to form the next population.

2.3.2 α-dominance strategy

While NSGA-II has been very successful, it measures solutions based on the Pareto dominance criterion, meaning that one solution outperforms another only if it is better in one objective and at least equal in other objectives. However, this strict Pareto dominance criterion can lead to some solutions remaining in the later generations, even when they exhibit extreme superiority in one objective and yet perform poorly in others. This issue arises due to the difficulty in optimising different objectives within GP, as the objectives are often in conflict, meaning that improving one objective can lead to a diminution of performance in another. As a result, these solutions are frequently generated in the early generations of GP and survive into the later generations. To overcome this, in our experiments, we will also use the α-dominance strategy [15], which relaxes the strict dominance criteria by introducing a parameter α that

controls the trade-offs among objectives. This strategy allows for weak trade-offs between objectives, providing a more flexible way to compare solutions. Equation 2.21 presents the α-dominance strategy:

$$G_i(A, B) = f_i(A) - f_i(B) + \sum_{j \neq i}^{1...m} \alpha_{ij}(f_j(A) - f_j(B)) \tag{2.21}$$

where $G_i(A, B)$ represents the evaluation of the comparison between solutions A and B with respect to an objective 'i', where $f_i(A)$ and $f_i(B)$ represent the values of objective 'i' in solutions A and B; $f_j(A)$ and $f_j(B)$ represent the values of objective 'j' in solutions A and B; and α_{ij} is the parameter controlling the trade-offs.

The key change here is that α-dominance considers not only the differences in objective values, but also the weighted trade-offs (α_{ij}) for each pair of objectives. When α_{ij} is set to 0, the traditional strict Pareto dominance is observed. On the other hand, by adjusting the values of α_{ij}, we can allow for weaker dominance relationships, capturing more diverse and potentially valuable solutions.

Choosing an appropriate α_{ij} is crucial and three adaptation schemes have been proposed [15] to determine suitable values. These adaptation schemes include linear, sigmoid, and cosine functions, each defined as follows:

$$\begin{aligned} f_{\text{linear}}(x) &= C - \frac{Cx}{50}, \\ f_{\text{sigmoid}}(x) &= C \cdot \frac{1}{1 + e^{(x-25)}}, \\ f_{\text{cosine}}(x) &= \frac{C}{2} \cdot \left(\cos\left(\frac{\pi x}{10}\right) + 1\right) \end{aligned} \tag{2.22}$$

where C is a sufficiently large constant.

In our work, we will use each one of these adaptation schemes. We will further discuss this in Section 6.2.2.

2.4 Conclusion

In this chapter, we have introduced the necessary background information for this thesis. First, we demonstrated traditional physical time series predictions and a novel event-based prediction in the field of financial forecasting. Next, we presented information on the framework of GP, including representation, population initialisation, genetic operators, breeding methods, and selection. This provides a comprehensive overview of GP. Finally, we introduced the multi-objective optimisation approach, NSGA-II, detailing its distinctive features. In the next chapter, we discuss the relevant literature of our work.

Chapter 3

Literature review

As presented in Section 2.1, financial forecasting is a way to predict the market's future movement. Because of the potential profit and unstable market, there has been extensive research into financial forecasting. To meet the growing need for accurate predictions in complex markets, algorithmic trading, particularly using machine learning algorithms, is widely employed. Therefore, in this literature review, we will discuss recent advancements and investigate gaps in the existing literature.

In the rest of this chapter, we first present the latest research of several well-known machine learning techniques in the field of physical time series prediction in Section 3.1, as this is the focus of Chapter 4. Section 3.2 discusses relevant works in the literature around the use of genetic programming – the primary algorithm used in this thesis – for financial forecasting. Further, in Section 3.3, we introduce works in the area of directional changes, which is the focus of chapter 5. Lastly, Section 3.4 reviews the machine learning application in the field of multi-objective optimisation, since MOO is the focus of Chapter 6.

3.1 Physical time series prediction

Physical time series prediction, as introduced in Section 2.1.1, is a prediction method based on fixed time series. Since most of the original data sets on the market are fixed time series, they are adopted quickly and widely. Predominantly, this research domain focuses on the

prediction of future movements within various financial time series. The scope of predictions encompasses a variety of targets, including stock price forecasting, index prediction, foreign exchange (forex) price prediction, commodity price forecasting, and volatility forecasting. Despite the variation in these forecasting targets, they share common underlying dynamics.

With the development of ML techniques, a significant evolution has occurred in the field of financial forecasting. ML techniques can quickly identify hidden features in large data volumes and are therefore able to achieve profits using a speed and frequency that is impossible for human traders. Since their introduction, various models have been built and applied to financial forecasting. ML applications in financial forecasting typically fall into two major categories:- price forecasting and directional movement forecasting. The extant research into price forecasting regards stock time series as being a predictable regression. The goal is to reveal the hidden regression within the historical data, thereby facilitating the prediction of future prices. Conversely, directional movement forecasting research aims to solve a classification problem in determining the potential trend of the current stock market.

In this section, we will introduce some popular ML algorithms used in this thesis and their applications in financial forecasting.

3.1.1 Gradient boosted machine

Gradient boosting machines (GBMs), as proposed by [39], represent an ensemble learning approach that iteratively refines predictions by incorporating new models to address the errors of preceding ones. This method has been employed for both classification and regression purposes. Specifically, as a classification tool, GBMs have been applied to the prediction of price movements, as evidenced by [40] and [41]. An enhanced variant of GBM, known as extreme gradient boosting (XGBoost), introduced by [42], has gained prominence in various data and ML competitions, as detailed by [43]. With outstanding RMSE and MAPE values, XGBoost has shown superior forecasting capabilities in the Forex market [44]. Additionally, the deployment of the XGBoost tree model in constructing a financial risk prediction framework has demonstrated significant accuracy and stability, outperforming alternatives such as support vector regression and back propagation (BP) neural network models, as discussed in

[45].

Moreover, GBM's applicability extends to price prediction, as illustrated by [46]. Its implementation with advanced deep learning methodologies, particularly long short-term memory (LSTM) networks, has been extensively researched, with notable contributions from [47] and [48]. A recent study by [49] innovatively employs critical features identified by XGBoost for the training of LSTM models, resulting in a predictive model that surpasses the performance of traditional autoregressive integrated moving average techniques in the Forex market.

3.1.2 Support vector machine

The introduction of the support vector machine (SVM) was made by [50]. As a kernel-based machine learning model, SVM has proven to be a powerful tool for solving classification and regression tasks. With its regression version, support vector regression, SVM become one of the most common ML algorithms applied in financial forecasting due to its solid theoretical foundations and strong generalisation capacity.

Most SVM applications focus on market movement direction forecasting, as demonstrated in studies such as [51] [52] [53]. Moreover, [54] explored the efficacy of SVM in conjunction with particle swarm optimisation (PSO) for cryptocurrency forecasting, revealing that SVM enhanced with PSO optimisation outperforms a normal SVM.

SVM has consistently been utilised for stock price predictions, with notable applications documented by [55] and [56]. Additionally, [57] employed SVM to forecast the closing prices of USD/TRY and EUR/TRY exchange rates, utilising varying kernel scale values, and found that the proposed model surpassed other algorithms in terms of performance. SVM's applications extend to addressing the challenges posed by high volatility in high-frequency data, as outlined by [58].

In more recent research, [59] discussed that a straightforward SVM model could achieve 60%-70% accuracy, with potential improvements when integrated with other algorithms such as random forest and genetic algorithm (GA).

3.1.3 Multi-layer perceptron

Multi-layer perceptron neural network (NN) is an NN with multiple interconnected layers [60], employing the back propagation algorithm for training the model. Notably simple in its architecture, the MLP has been extensively utilised within the domain of financial forecasting, serving primarily as a classifier, as evidenced by research conducted by [61] and [62].

The Forex market generates large amounts of high-frequency data every minute, whereas traders tend to make decisions over longer periods of days, weeks, months, or quarters. Therefore, [63] implemented interval time series[1] combined with interval MLP, an advanced version of MLP introduced by [64]. This approach integrates interval data, resulting in outcomes that show promise compared to the interval random walk.

In innovative research, [65] proposed a novel algorithm containing MLP tuning by an improved GA. The four main parameters, namely network depth, network width, density layer activation function, and network optimiser, were tuned by the GA. The empirical findings of this exploration underscore the hybrid model's superiority over traditional ML counterparts in achieving higher predictive precision. Other related works containing evolution algorithms could be found in[66] [67].

Moreover, the application of MLP is not confined to classification challenges but extends to regression problems, as illustrated by [68], [69], and [70]. An intriguing adaptation, the evolving multi-layer perceptron (EMLP), which merges evolving connectionist systems with MLP, was employed by [71] for the precise forecasting of gold prices. This methodology, which leverages daily price fluctuations and a comprehensive dataset, has demonstrated exceptional accuracy, as evidenced by its MAPE metrics.

3.1.4 K-nearest neighbours

The K-nearest neighbours (KNN) algorithm is one of the prominent supervised learning methods used in both classification and regression. First introduced by Fix and Hodges Jr. in 1951 [72], it makes forecasts based simply on the nearest neighbours to a given data point. Despite

[1]Interval time series assign an interval of values at each period, such as daily or monthly minimum and maximum values.

its simplicity, KNN has been successfully applied in time series forecasting.

In the field of financial forecasting, KNN has garnered increased preference due to its adaptability and effectiveness. [73] applied principal components analysis to overcome the high calculation requirements and reduce the redundant information inherent in the KNN. [74] implemented the KNN algorithm for the prediction of monthly gold prices, illustrating its applicability in commodity markets. [75] presented a novel approach combining LSTM and KNN in the Forex market. Furthermore, [76] developed a complex approach combining empirical mode decomposition and KNN to predict the opening price and closing price of the stocks. [77] proposed an improved KNN approach that considers the price trend of several previous trading days, showing precise results in short-term prediction.

More relative works could be found in [78, 79, 80, 81].

3.1.5 Decision tree

A decision tree is one of the simplest ML algorithms which comprises a tree-like structure. As a top-down modality, a decision tree is built from the root node to the internal node and ends with the terminal node [82]. Through these nodes, data is divided into smaller subsets with similar values.

Despite its simplicity, the decision tree performs effectively and is frequently used as a benchmark in comparisons with other ML algorithms in financial forecasting [83] [84]. For instance, [85] evaluated the performance of four algorithms—neural network, decision tree, naive Bayes, and K-nearest neighbours—for future market prediction, finding that a decision tree model with three classes was the most effective. Additionally, [86] compared the decision tree and support vector Regression to develop a model for gold price prediction, demonstrating that the decision tree achieved lower mean square error and faster computation. More recently, [87] combined traditional data analysis and sentiment analysis to predict future market trends. The decision tree incorporating sentiment analysis was shown to have more than 90% accuracy on real-world datasets.

3.1.6 Random forest

Random forests (RFs) [88], based on the bagging technique over decision trees, have become an efficient and extensively used statistical learning algorithm, gathering attention to both classification and regression problems.

The application of RFs in forecasting market movements is well-documented, with significant contributions made by [89] and [90]. In an innovative approach, [91] transformed financial forecasting into a four-class categorisation, introducing categories for 'strong positive' and 'strong negative' movements. These additional signals were identified as a key factor in enhancing predictive capability. Further, [92] explored an ensemble technique that integrates RF with XGBoost, aiming to ascertain the directional movement of stock prices from a pre-defined point in time. The effectiveness of this model was corroborated across a range of metrics, underscoring its robustness.

In the realm of regression prediction, RF has demonstrated commendable performance, as evidenced by studies such as [93] and [94]. An extensive comparison by [95] among various regression models—including linear regression, SVR, decision tree, RF, and extra tree regression—revealed that RF and decision tree models exhibited superior outcomes.

Due to its superior performance, RF has been identified as a tool to advance deep learning models. Specifically, [96] proposed a novel integration of LSTM and RF to address the overfitting issue commonly associated with LSTM models. This hybrid model not only delivered stable performance, but also offered insights through variable importance analysis, enhancing interpretability. Additional research employing RF to enhance forecasting models includes [97], [98], and [99], illustrating the potential of RF in improving the predictive accuracy and reliability of financial forecasting models.

3.1.7 Extra trees

Extra trees (ETs), or extremely randomised trees, represents an advancement in tree-based ensemble ML models, designed as an extension of the RF framework to overcome overfitting issues, as elucidated in [100].

This model has demonstrated efficacy in addressing regression challenges, as highlighted in [95]. In [101], a variety of regression models were assessed, including the CatBoost regressor, gradient boosting regressor, ET regressor, AdaBoost regressor, K-nearest neighbour regressor, and the Theil-Sen regressor. The findings from this examination revealed that the ET model exhibited the highest MAE value, signifying its prominent performance in the context of Bitcoin price prediction.

Furthermore, an exploration of the existing literature reveals the frequent application of ET within ensemble models for classification tasks [102]. [103] compare the effectiveness of tree-based ensemble ML models, including RF, XGBoost, Bagging Classifier (BC), AdaBoost Classifier (Ada), ET, and Voting Classifier (VC) in forecasting the direction of stock price movements. Additionally, the Kendall W test was employed to rank these tree-based models. It was determined that the ETs classifier surpassed the performance of its counterparts.

3.2 Genetic programming (GP)

GP was initially introduced by [29] and has since been used in solving financial forecasting problems for over 20 years [8]. Due to its evolutionary process and symbolic regression, GP can generate profitable and human-readable models. Consequently, GP has been employed to address various issues in financial forecasting. Many studies focus on forecasting in different markets. For instance, [104] used GP to establish technical trading rules in the Forex market. Similarly, [105] investigated the predictability of GP in the Forex market, constructing trading rules based on predicted values. Their findings indicated that GP provided a profitable trading strategy with modest predictability for one-day-ahead forecasting, though this predictability disappeared when forecasting further ahead. In addition, the profitability of the models disappeared after taking transaction costs into account. Some other works of the GP application in the Forex market could be found in [106] [107].

GP has also been applied to the stock market. For example, [108] presented GP-based technical trading rules that were able to outperform the buy-and-hold trading strategy on S&P500 when taking into account transaction costs. Other notable works include [109] and

[110].

Further, the application of GP could use data from various periods. [111] conducted research using daily data, while [112] adopted high frequency data at millisecond intervals. In [10], GP was compared with traditional forecasting devices across different data periods, including yearly, quarterly, monthly, and others.

GP has demonstrated competitiveness against other algorithms. For example,[113] built a 1-day-ahead trading system based on GP, allowing them to investigate stock trades made by groups of artificial traders. The result showed the GP-based system dominated several benchmark models, including the random walk model and the buy-and-hold strategy, at least in terms of short-term predictions. Besides, GP was also compared with a NN, another popular ML model, and again revealed its effectiveness[13]. More recently, [11] created an automated system, one that combines multi-objective optimisation, GP, technical analysis, and feature selection. They evaluated the performance of the system in six BOVESPA[2] shares for two periods, from 2013 to 2015 and from 2015 to 2016. The system makes a profit even when the asset is devalued. Moreover, [10] provided evidence that the GP system is competitive against traditional algorithms such as the auto-regressive integrated moving average and exponential smoothing *(in some cases with statistical significance)*. Another common and successful application of GP is under directional changes[114]. These authors provided evidence that the new approach, when combined with GP and directional changes, was able to generate new and profitable trading strategies.

Lastly, technical analysis, particularly through the use of technical indicators, is able to enhance the prediction performance of GP by focusing on the potential features and minimising market errors. [115] presented a GP-based system that incorporates six well-known technical indicators, showing better performance than the individual technical rules. More recently, [116] used five trend-following indicators as the input of the GP to develop the trading rules on daily Bitcoin historical prices, outperforming the buy-and-hold strategy. Another recent GP application is [117], which combined indicators from technical and sentiment analysis.

[2]BOVESPA stands for the São Paulo Stock Exchange and is the largest stock exchange in Brazil.

3.3 Directional changes

Directional changes (DCs), originally introduced by [14] and defined formally by [9], is a novel approach summarising fixed time series into event-based series. Although physical time series predictions yield results in the financial forecasting domain, it nonetheless has the drawback of including a discontinuous time series. Instead of analysing historical data at fixed intervals, event-based prediction focuses on significant events in the market. A significant event is defined and identified as the price movement over a user-defined threshold, while movements below this threshold are considered to be irrelevant information and noise. As a relatively new method, DC has demonstrated success in financial forecasting. This section will introduce the state-of-the-art application of DCs.

DCs have gained traction in handling non-constant time intervals and high-frequency data, notably in tick data derived from the Forex market, as evidenced in works such as [118, 119, 120, 114, 121]. A notable DC-based trading strategy, ZI-DC0, was developed by [122] using trend-following and contrary trading technical indicators. Based on the ZI-DC0, [123] proposed an improvement, and ZI-DC1 demonstrated improved performance. Additionally, [124] introduced an automated system, DCT2, one that can observe price changes in the market and adjust dynamically.

There are two major aspects of the DC framework. Firstly, the forecasting of the directional changes (DC) and overshoot (OS) events holds significant importance for maximising profits. Some empirical scaling laws[3] under the DC framework were proposed as [125] and [28] defined empirical scaling laws to quantitatively connect price movements and transactions within the Forex market. Building on this work, [126] extended it with additional scaling laws, yielding positive outcomes in the Forex market. One notable scaling law indicates that, on average, an OS event takes twice as long as the DC event to achieve the same price change. These empirical scaling laid the foundation for the later research [126].

For example, [127] introduced DC-based trading strategies based on the average overshoot length scaling law proposed by [125], resulting in positive returns across various stock

[3] Scaling laws establish the mathematical relationship of how certain properties of financial markets change with different scales or thresholds.

market indices. [128] identified three independent variables, all of which were relevant to predicting OS event endpoints. Moreover, they formulated a forecasting problem of whether the next OS event endpoint value would exceed a user-defined threshold. [129] proposed one single independent variable to answer the question of whether the current trend will continue for a specific percentage before the trend ends. Subsequently, [130] proposed a dynamic threshold definition method that outperformed other algorithms using a fixed threshold. In [131], they applied their finding to detect the events under the DC framework.

Following this, [132] proposed a novel DC trading strategy challenging the assumption that DC events are always succeeded by OS events, thereby demonstrating a superior performance as compared to other DC-based strategies and the buy-and-hold approach across 20 Forex currency pairs. A recent study, [133] utilised genetic algorithms to optimise multiple DC-based trading strategies using data from 20 Forex markets, outperforming single DC-based strategies, traditional technical indicators, and the buy-and-hold approach.

The second significant aspect revolves around the application of technical analysis within the DC framework. Traditionally, technical analysis has been applied in physical time, primarily employing technical indicators, revealing hidden features for algorithms. Like physical time, technical analysis also could be applied under the DC framework. [5] introduced the first set of DC-specific indicators, with additional DC indicators proposed in [6] and [134]. More recently, [135] used the average true range indicator to dynamically determine the DC thresholds. [136] used both DC-based indicators and scaling laws under the DC framework to construct trading strategies.

3.4 Multi-objective optimisation

In 2008, [137] developed a MOO algorithm containing particle swarm optimisation using end-of-day data to optimise two objectives, namely the Sharpe ratio and per cent profit. This model exhibited superior performance when compared to five distinct technical indicators. Subsequently, Butler and Daniyal attempted to make accurate predictions on the movements of the stock market. To achieve their goal, they constructed an evolutionary artificial neural

network (ANN) trained by 23-year data. Their findings revealed that the MOO approach yielded higher investment returns than the Single-Objective Optimisation (SOO) methods [138].

Further advancing this field, Lohpetch and Corne [139] observed the robust performance of the buy-and-hold strategy attributed to its infrequent trading approach. They proposed a MOO strategy that outperformed not only the buy-and-hold approach, but also the SOO methods, particularly by employing a more frequent decision-making strategy.

[140] proposed a differential evolution model analysing four technical indicators to optimise profit, risk and the number of trades. To do so, they used the MOO procedure to generate Pareto fronts for each indicator. The research took place over a 10-year period, using daily data from the IBOVESPA index. By benchmarking the default values of the respective technical indicators used and the buy-and-hold strategy, the proposed differential evolution algorithm was shown to have superior performance.

In addressing the inadequacy of optimal technical indicators for complex market movements in the real world, [141] proposed a rule change trading system using a GA to maximise two objectives, specifically the pay-off ratio and profit. The proposed system showed a complete result compared with the system not using GA and the buy-and-hold strategy.

More recently, [142] conducted an in-depth study regarding the performance of the decision model on three of the state-of-the-art dynamic multi-objective optimisation algorithms[4]. They evaluated dynamic vector-evaluated particle swarm optimisation, multi-objective particle swarm optimisation with crowded distance, and dynamic NSGA-II in the Forex market. The result revealed that the dynamic NSGA-II was the most stable algorithm. In addition, [143] developed a new model based on support vector regression with a wrapper-based feature selection approach employing MOO. The primary aim of their work was to forecast future prices of crude oil, demonstrating the potential of this advanced methodology in predictive analysis.

Lastly, it should be noted that to the best of our knowledge, no DC literature currently uses MOO.

[4]Dynamic multi-objective optimization algorithms aim to determine changes in the Pareto front and respond accordingly throughout the algorithm's process.

3.5 Conclusion

From the investigation of the literature on physical time series prediction in Section 3.1, we can conclude that ML algorithms have been widely adopted to solve time series forecasting problems in finance. However, due to the unpredictability and complexity of the stock market, there was not one ML algorithm that showed significant and outstanding performance. From Section 3.2, we can see that GP has been used successfully in financial forecasting. However, experiments take place over a limited number of markets (usually 1-2), e.g., [108]. In general, GP has only been compared with a few benchmark algorithms, usually less than 6 (e.g., [13] and [10]). It is also worth noting that some works took transaction costs into account, but others did not. In addition, there is a great deal of variation in terms of stock markets, data periods, and forecasting days ahead in existing published works, making any form of comparison between various published works very challenging. Therefore, one of our goals in this thesis is to address these gaps by considering all the aforementioned factors and thoroughly investigating a financial GP algorithm. In Chapter 4, we will investigate the performance of a GP algorithm and benchmark it against 9 ML algorithms taking transaction costs into account within our trading strategy.

In Section 3.3, the literature on DC was introduced. As a novel method, DC has achieved notable performance in financial forecasting. However, the majority of existing literature under the DC framework has focused on the Forex market. There is however relatively limited research on the application of DC indicators in the stock market. Specifically, no previous work has successfully predicted future stock movement using a large number of DC indicators. Furthermore, exploiting indicators from the DC paradigm for constructing trading strategies is still largely unexplored compared to the technical analysis in the physical time domain. Given that GP did a great job in combining diverse indicators for profitable trading strategies, as observed in [117, 144, 145, 2], it is natural to explore the potential of using DC-based indicators as part of a GP trading algorithm. We are thus investigating using the DC-based indicators as part of the GP introduced in Chapter 4. This will allow us to investigate the advantages and disadvantages of using DC, as outlined in Chapter 5.

Finally, Section 3.4 discussed the effectiveness of MOO in predicting stock market movements and revealed the research gap of limited studies considering multi-objective approaches under the DC framework. MOO will form the basis of Chapter 6, in which we will explore the advantages of using MOO under a DC framework against a single objective GP algorithm.

Chapter 4

Genetic programming application on physical time scale

4.1 Motivation

This chapter introduces the first contribution of this thesis, and we will focus on physical-time data. As mentioned in Section 3.1, technical analysis is a very commonly used technique under the physical-time paradigm. ML algorithms have been successfully adopted in many real-world applications, including financial forecasting [108] [113] [10] [11]. Such algorithms tend to use technical indicators and report results under both high-frequency (e.g., minute-by-minute, hourly, etc.) and low-frequency (e.g., daily) data [146] [147][148][149]. ML is favoured by traders because it can effectively discover patterns in historical price data over different time periods. GP is one of the most common ML algorithms applied to financial forecasting due to its ability to perform global search in terms of both exploration and exploitation, as well its capacity to produce white box models.

However, as mentioned in Chapter 3.5, most published works in this domain tend to examine and compare GP with only a small selection of ML algorithms (usually less than four, e.g., [13] and [10]). Additionally, such published works tend to focus on datasets from a limited number of international markets (usually one or two, e.g., [108]). It is also worth noting that some works do not even take transaction costs into account, which can inflate

their results and create a misleading impression of their real-world performance.

Therefore, our goal in this chapter is to fill the gap in the literature by considering all the aforementioned factors and conducting an in-depth investigation of a financial GP algorithm. To achieve this goal, we apply a GP-based trading strategy and compare it with nine popular ML algorithms, namely gradient boost (GB), stochastic gradient descent (SGD), random forest (RF), multi-layer perceptron (MLP), extra tree (ET), passive aggressive classifier (PAC), C-support vector classification (SVC), k-nearest neighbors (KNN), and decision tree (DT). The experiments take place over 100 datasets from 10 international markets. Additionally, we investigate the effect of using different period lengths for training/testing purposes, using both a 5-year and a 10-year period. This essentially doubles the number of datasets from 110 to 220. The reason behind this investigation is that many references in the ML literature indicate that more data can allow an ML algorithm to better generalize. We also summarize our results in terms of different stock markets and countries, in an attempt to identify more strongly performing markets. Finally, we compare the GP with a traditional financial benchmark (i.e., buy and hold). Our overall aim is to conduct an in-depth analysis of the performance of different ML algorithms and report on how various factors (such as the period length and financial market) can affect the algorithms' financial respective performance. Transaction costs are taken into account for each trading action an algorithm performs.

The rest of this chapter is organised as follows. Our proposed methodology is provided in Section 4.2, in which we first present the details of the GP and then discuss how the GP models are used as part of a trading strategy. Section 4.3 provides a description of our experimental setup and presents the datasets used in our experiments, together with the benchmarks and the parameter tuning process. Then, in Section 4.4, we discuss the results of our experiments. Finally, Section 4.5 concludes this chapter by highlighting its major findings.

4.2 Methodology

In this section, we present in detail the physical-time GP algorithm used in this chapter. We start by discussing in Section 4.2.1 important GP components, such as the terminal and the

function sets, the representation of the GP individuals, and the selection method, along with the genetic operators used. Then, in Section 4.2.2 we present an overview of the trading strategy that the GP employs and conclude by how it is evaluated (Section 4.2.3).

4.2.1 Genetic programming model

4.2.1.1 Terminal set

Here we employ 146 different technical indicators as terminals in our GP trees. All the technical indicators that are used in this work are as listed in Table 4.1. We selected 5, 10, 15, and 30 days as the periods for most indicators. For those indicators that require two periods, we have selected pairs of periods as [5,10], [5,15], [10,15], [10,30], and [15,30]. It is worth noting that few indicators do not need periods, e.g., On Balance Volume, Ease of Movement, and Market Facilitation Index. We introduced some indicators earlier in Chapter 2.1.2. Since technical indicators are not the primary focus of the thesis, the remaining indicators, alongside their respective formula, are provided in Appendix A.

In conjunction with these indicators, we employ an ephemeral random constant (ERC) which, upon invocation, generates a random number distributed uniformly between -1 and 1. To ensure consistency with ERC's range, the DC indicators have been appropriately normalised.

4.2.1.2 Function set

The function set includes two logical operators, namely AND and OR. It also includes two logical expressions, namely less than ($<$) and greater than ($>$).

4.2.1.3 Model representation

The GP evolves logical expressions using operators like AND, OR, $<$, or $>$. These expressions become the first branch of an if-then-else (ITE) statement, as shown in Part 1 of Figure 4.1. The ITE tree also includes a 'then' branch, which stands for a buy action, and always returns a value of 1, and an 'else' branch, which represents a hold action and consistently returns a

Table 4.1: Technical indicators and the corresponding periods used in this thesis.

Categories	**Indicators**	**Periods (days)**
Market Strength Indicators	Money Flow Index (MFI) [Equation A38]	5,10,15,30
	Accumulation/Distribution (A/D) [Equation A21]	
	On Balance Volume (OBV) [Equation 2.10]	
Momentum Indicators	Momentum (MTM) [Equation A2]	(5,10,15,30)
	Relative Difference in Percentage (RDP) [Equation A3]	(5,10,15,30)
	Rate of Change (ROC) [Equation A4]	(5,10,15,30)
	Disparity index [Equation A5]	(5,10,15,30)
	Percentage Price Oscillator (PPO) [Equation A6]	([5,10],[5,15],[10,15],[10,30],[15,30])
	Ease of Movement (EOM) [Equation A30]	
	Stochastic Momentum Index (SMI) [Equation A51]	(5,10,15,30)
	Vertical Horizontal Filter (VHF) [Equation A56]	(5,10,15,30)
Volatility Indicators	Average True Range (ATR) [Equation 2.7]	(5,10,15,30)
	Relative Volatility Index (RVI) [Equation A16]	(5,10,15,30)
Oscillating Indicators	Relative Strength Index (RSI) [Equation 2.1]	(5,10,15,30)
	Relative Momentum Index (RMI) [Equation A46]	([5,10],[5,15],[10,15],[10,30],[15,30])
	Stochastic Oscillator (K% and D%) [Chapter 8]	(5,10,15,30)
	Commodity Channel Index (CCI) [Equation 2.3]	(5,10,15,30)
	Williams' %R [Section 2.1.2]	(5,10,15,30)
	Chande Momentum Oscillator (CMO) [Equation A9]	(5,10,15,30)
	De-trended Price Oscillator (DPO) [Equation A22]	(5,10,15,30)
	Klinger Oscillator (KO) [Equation A32]	
	Mass Index (MI) [Equation A39]	
	Percentage Volume Oscillator (PVO) [Equation A41]	([5,10],[5,15],[10,15],[10,30],[15,30])
Trend Indicators	Moving average (MA) [Equation 2.2]	(5,10,15,30)
	Exponential Moving Average (EMA) [Equation 2.9]	(5,10,15,30)
	Double Exponential Moving Average (DEMA) [Equation A23]	(5,10,15,30)
	triple exponential average (TRIX) [Equation A54]	(5,10,15,30)
	Volume Adjusted Moving Average (VAMA) [Equation A57]	(5,10,15,30)
	Moving Average Convergence/Divergence (MACD) [Equation A58]	([5,10],[5,15],[10,15],[10,30],[15,30])
	Aroon Indicator (up and down) [Equation A7 and A8]	(5,10,15,30)
	Donchian Channels [Equation A15]	(5,10,15,30)
	Directional Movement Index (DMI) [Equation A24]	(5,10,15,30)
Other indicators	Market Facilitation Index (MF) [Equation A11]	
	Negative Volume Index (NVI) [Equation A40]	
	Parabolic SAR [Equation 2.11]	
	Polarised Fractal Efficiency indicator (PFE) [Equation A42]	([5,10],[5,15],[10,15],[10,30],[15,30])
	Random Walk Index (High and low) [Equation A44 and A45]	(5,10,15,30)

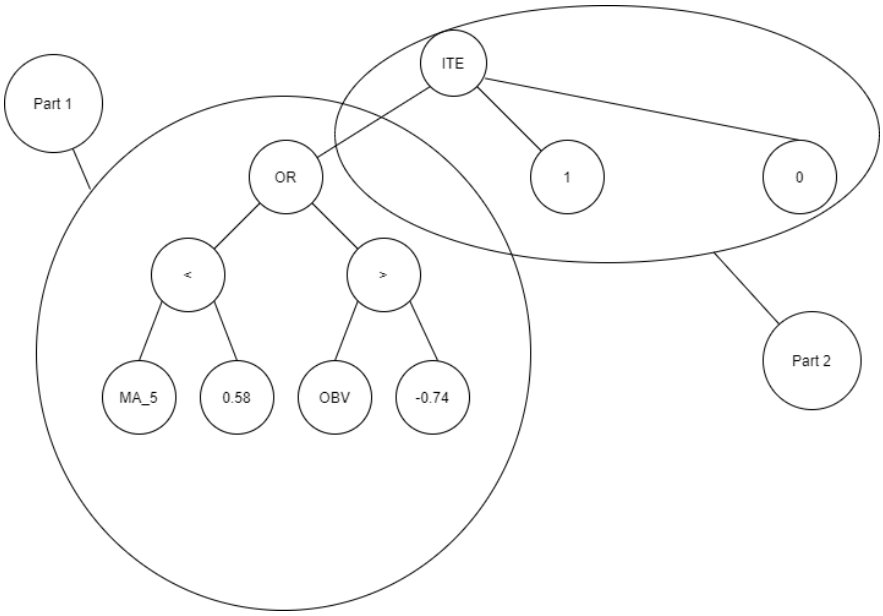


Figure 4.1: Example of the GP tree structure and the if-then-else structure

Table 4.2: Configuration of the GP algorithm

Configuration	Value
Function set	AND, OR, $>$, $<$
Terminal set	146 technical indicators
Genetic operators	Elitism, subtree crossover and point mutation
Selection	Tournament

value of 0. Note that this structure does not cover a sell action, and we will provide more details on that in Section 4.2.2. Part 2 of the GP is not involved in evolution because it deals with constant values (0 or 1). The choice between the 'then' branch (indicating a buy signal) and the 'else' branch (indicating a hold signal) depends on whether the GP individual's expression evaluates to True or False.

4.2.1.4 Selection method

In our framework, we employ elitism, sub-tree crossover, and point mutation as genetic operators. Furthermore, we utilise tournament selection to determine the parental candidates for the aforementioned genetic operators [150]. An overview of the GP configuration is provided in Table 4.2.

4.2.2 Trading strategy

The objective of a GP individual, representing a trading strategy, is to predict whether the price will increase by a given percentage, denoted as $r\%$, within a predefined time frame of n days. If the GP individual generates a 'True' prediction, we initiate a purchase of a single unit of the stock unless we currently hold it. Conversely, if the prediction is 'False', we maintain our current position (hold). In instances where we already possess the stock and its price rises by at least $r\%$ within the subsequent n days, we execute a sale on the day of the increase. If the price fails to increase by the defined threshold within the specified time frame, we sell the stock on the n-th day *(note that our trading strategy does not involve short-selling)*. After each selling action, we compute the resulting profit, incorporating transaction costs set at 0.025% per trade. The summarisation of our trading strategy is outlined in Algorithm 3.

Algorithm 3 Our trading strategy with input the threshold $r\%$ and the number of days n

Require: O is the outcome of the GP tree and $\mathit{flag} \in \{0, 1\}$ indicates whether we already hold the stock (i.e., 1) or not (i.e., 0)

if $O = 1$ and $\mathit{flag} = 0$ **then**
    Buy one share of stock and set $\mathit{flag} \leftarrow 1$
    $t_0 \leftarrow t$ //Mark the time of trade: t is used for the current time
    $p_0 \leftarrow p$ //Mart the price when buying: p is used for the current price
else
    if ($\mathit{flag} = 1$ and $p > (1 + r/100) \times p_0$) OR $(t - t_0) > n$ **then**
        Close the trade and set $\mathit{flag} \leftarrow 0$
        Calculate the resulting profit
    end if
end if

To assess the trading strategy's performance for a specific stock, we rely on two key metrics, specifically the expected rate of return ($\mathbb{E}[RoR]$) and risk. The rate of return (RoR) is a measure of how profitable a particular trade turned out to be. It is a particularly useful metric for short-term investors, as it allows them to evaluate individual trades. The expectation $\mathbb{E}[RoR]$ over all such events in a given period (e.g. the training or test periods) is, therefore, a measure of how profitable trades were for that period on average, given a particular trading strategy. The second metric, risk, serves as an indicator of the uncertainty and potential for financial loss associated with the fluctuation of the RoR, as expressed in Equa-

tion 4.3. Additionally, we use total return (TR) to assess the overall efficacy of the algorithm when comparing the GP-based algorithm with the buy-and-hold strategy, as the risk metric requires multiple trades, whereas the buy-and-hold strategy has only one trade through the whole trading period. Because buy-and-hold makes only a single trade (while the GP-based algorithm makes several), it is fairer to compare them across the TR rather than also using the $\mathbb{E}[RoR]$. Furthermore, TR's advantage over the $\mathbb{E}[RoR]$ as a metric is that, in the presence of more trade events having a similar RoR, the TR will be higher, accurately reflecting the fact that the trades during that period resulted in better overall profit. TR is calculated by aggregating the returns listed in the RoR dataset and then dividing this sum by the initial cost incurred during the first transaction, as formalised in Equation 4.1. Among the three metrics, the TR and $\mathbb{E}[RoR]$ are maximisation objectives, and risk is a minimisation one.

$$TR = \frac{\sum_{i \in \mathsf{RoR}} \Big[(1-c) \cdot P_s(i) - (1+c) \cdot P_b(i) \Big]}{(1+c) \cdot P_b(i_0)} \cdot 100\% \tag{4.1}$$

$$\mathsf{RoR}(n) = \frac{(1-c) \cdot P_s(i) - (1+c) \cdot P_b(i)}{(1+c) \cdot P_b(i)} \cdot 100\% \tag{4.2}$$

$$\mathsf{Risk} = \sqrt{\mathsf{Var}(\mathsf{RoR})} \tag{4.3}$$

where the i indices here correspond to completed trade events (i.e. where both a buy and a sell event have taken place for a particular asset) within the period of interest; $P_s(i)$ refers to the sell price for that event; $P_b(i)$ refers to the buy price for that event; $P_b(i_0)$ denotes the buy price of the first ever trade event for that period; and, c is the transaction cost.

4.2.3 Fitness function

We use the Sharpe ratio, defined in Equation 4.4, as the fitness function for the proposed algorithms. That is, each algorithm's objective is to maximise the Sharpe ratio which takes into account both returns and risk.

$$\text{Sharpe ratio} = \frac{\mathbb{E}[RoR] - R_f}{\text{Risk}}, \tag{4.4}$$

where $\mathbb{E}[RoR]$ stands for the sample mean of the list of the rate of returns for a given stock and R_f is the risk-free rate.

In this way, we optimise trading strategies based on a comprehensive metric that considers both returns and risk rather than focusing solely on one aspect.

4.3 Experimental set-up

In this section, we introduce the financial data employed and present the benchmarks used for the experimental comparison. Further, we discuss the parameter-tuning process for the GP and the associated trading strategy.

4.3.1 Data

As discussed before, our goals in this Chapter can be summarised as follows: (i) Compare the performance of the GP algorithm against 9 popular ML algorithms; (ii) investigate the effects of using a longer period (10 years) against a shorter period (5 years); (iii) identify those markets and countries that perform better than others; and, (iv) compare the GP performance against the buy-and-hold benchmark.

In order to achieve the above, we used daily data for 110 stocks derived from 10 markets across 6 countries. These markets are the Dow Jones Industrial Average (DJIA); the Nasdaq Stock Market (NASDAQ); the New York Stock Exchange (NYSE); the Russell 2000 Index; and the Standard and Poor 500 (S&P500) in the United States, in addition to the Nifty Fifty (NIFTY 50) in India; the Taiwan Stock Exchange Corporation (TSEC); the DAX performance index in Germany; the Nikkei 225 in Japan, and the Financial Times Stock Exchange 100 Index (FTSE 100) in the United Kingdom. To investigate the effects of using a longer training and test period, we use two sets of periods, specifically of 5 and 10 years. Thus, the selected 110 stocks are first examined over the period from 2015-2020 (5 years) and then for the period 2010-2020 (10 years). While fundamental statistical principles dictate that more data

leads to improved statistical efficiency, there is, however, also evidence that old data might be irrelevant to solving financial problems, e.g., [12]. Therefore, since we used two different periods to train and test the algorithms, we ended up with 220 datasets. We divided each period into a training, validation and test set with the following ratio 60%:20%:20%. The validation set is used for parameter tuning.

Before conducting the experiments, we first cleaned the data, including removing incorrect, missing, and null values from each dataset. Furthermore, these data were converted into technical indicators as presented in the previous sections. Data normalisation was also performed to set indicator values to between -1 to 1.

4.3.2 Benchmarks

In order to evaluate the performance of GP, we need to compare it with some benchmarks. As noted earlier, we have selected nine ML algorithms, namely GB, SGD, RF, MLP, ET, PAC, SVC, KNN, and DT. We use the above algorithms to tackle a binary classification problem in the form of 'Is the stock price going to increase by $r\%$ within the next n days?' Class 1 denotes a buy action, while Class 0 denotes a hold action. The sell action takes place again as a part of the trading strategy that was described earlier in Section 4.2.2. All nine ML algorithms are implemented via Scikit-learn library on Python.

Apart from that, we compare proposed algorithms against the buy-and-hold strategy, a trading strategy wherein traders buy a stock and hold it for a long time with the hope of an increase over a long period [151]. Buy-and-hold allows us to compare the GP-based algorithms' performance against the market. However, the net profit in buy-and-hold is the difference between the price at the beginning and the end of the trading period. This means that there is only one buy option and one sell option that occurred throughout the entire trading period for each data set, and this causes issues when evaluating using $\mathbb{E}[RoR]$ and risk. Therefore, we apply TR as the metric when comparing the proposed model and the buy-and-hold strategy in Section 4.4.4.

Table 4.3: Selected parameters of the GP algorithm after parameter tuning

Parameter	Value
Max depth	6
Population size	500
Crossover probability	0.95
Crossover probability	0.05
Tournament size	2
Numbers of generation	50

4.3.3 Parameter tuning

We conducted a grid search to determine the optimal GP parameters for three GP-based algorithms, with parameter tuning carried out using the validation set. Table 4.3 presents the selected parameters and their corresponding values after tuning.

For the trading strategy, we need to decide two parameter values from the question, 'Will the price increase by at least $r\%$ in the next n days?' to make sense of our findings.

Instead of globally tuning these parameters to come up with the best set of values across all datasets (as performed for the GP), we opted for customised values for each individual dataset. This approach aimed to maximise the performance of the trading strategy on a per-dataset basis. These three parameters take values from the following space:

- n (prediction horizon): 1, 5, 15
- r (price movement percentage): 1%, 5%, 10%, 20%

4.4 Results and analysis

The analysis is divided into four parts. In the first part, Section 4.4.1, we present the GP results and compare them against the 9 selected ML algorithms. In Section 4.4.2, we study the GP's performance across different financial markets and countries. In Section 4.4.3, we present the results from the comparison between 5 and 10 years' worth of data. Finally, in Section 4.4.4, we present the buy-and-hold results and compare them against the GP's results. All results are presented in terms of two financial metrics, namely expected rate

Table 4.4: Average (standard deviation) $\mathbb{E}[RoR]$ and risk results of GP and other ML algorithms. Best value per metric is shown in boldface.

Algorithms	**Expected Rate of Return**	**Risk**
GP	0.2376%(0.0678)	0.0395 (**0.0632**)
DT	0.7601% (0.0522)	0.1126 (0.0952)
ET	1.3826% (0.1039)	0.0663 (0.0848)
GB	1.2284% (0.0665)	0.1054 (0.0958)
KNN	1.0568% (**0.0393**)	0.0771 (0.0784)
MLP	1.0889% (0.0605)	0.0942 (0.0766)
PAC	1.9016% (0.0738)	0.0974 (0.0817)
RF	1.6348% (0.1162)	0.0768 (0.0839)
SGD	**1.9761%** (0.0613)	0.0864 (0.0992)
SVC	1.5736% (0.0648)	**0.0275** (0.0729)

of return ($\mathbb{E}[RoR]$) and risk. All the metrics are calculated by the average performance of 50 independent runs. To examine the statistical significance of each section's results, we performed the non-parametric Kolmogorov-Smirnov (KS) test.

4.4.1 GP vs ML algorithms

The analysis commences with a comparison of the performance of GP with 9 common classification models. Table 4.4 presents the average (standard deviation) $\mathbb{E}[RoR]$ and risk on GP and 9 classification algorithms. From Table 4.4, we can observe that, in terms of expected rate of return, all algorithms have yielded positive returns, with SGD having the highest return at 1.97621% per trade. The lowest return is delivered by the GP at 0.2376% per trade. In terms of risk, the lowest risk comes from SVC (0.0275), while the GP has the second lowest risk at 0.0395. By looking at standard deviation values, GP and 9 classification algorithms have similar values in terms of expected rate of return and risk, where GP and KNN achieved the best standard deviation values.

Additionally, we performed the non-parametric KS test for GP and other ML algorithms. We used GP as the control algorithm, and each KS test compares the GP's distribution against a different ML algorithm's distribution. The null hypothesis is that the two distributions come from the same continuous distribution. To account for the nine multiple comparisons (nine ML algorithms compared to the GP), we apply the Holm-Bonferroni correction. Consequently,

the minimum acceptable p-value for achieving statistical significance at a 5% level is determined by Equation 4.5.

$$\alpha_{\text{rank}} = \frac{\alpha}{m + 1 - \text{rank}} \tag{4.5}$$

where $alpha = 0.05$ for the 5% significance level, $m = 9$, since we have 9 multiple comparisons, and rank $\in$ {1, 2, 3, 4, 5, 6, 7, 8, 9}, and 9-rank+1, which varies for different ranks of the p-values found. The rank denotes the order of magnitude of the p-values, with 1 representing the smallest and 9 the largest. The ranked p-values reveal any significant differences arising between the samples, wherein the first p-value should be less than 0.0056, the second less than 0.0063, the third less than 0.0071, the fourth less than 0.0083, the fifth less than 0.01, the sixth less than 0.0125, the seventh less than 0.0166, the eighth less than 0.025, and the ninth less than 0.05.

Table 4.5 presents the results of the KS test. When a difference is statistically significant at the 5% level, this is indicated by putting the relevant p-value in boldface. As we can observe, in terms of expected RoR, there are no statistically significant differences apart from the pairing of GP and SVC. So, even though the mean expected RoR of the GP was lower when compared to the other ML algorithms' mean expected RoR, *this difference was not statistically significant at a 5% level*.

In terms of risk, we can observe that all comparisons of the distribution pairs are statistically significant. Given that the GP ranked second best, this indicates that the GP statistically outperformed DT, ET, GB, KNN, MLP, PAC, RF, and SGD, while it was statistically outperformed by SVC.

4.4.2 Market and countries

In this section, we analyse the results in terms of different financial indices and countries to investigate if there are any particular markets with stronger performance. In particular, we delve into the GP results, given its competitive performance from the previous section.

Table 4.6 shows that there are six indices with positive $\mathbb{E}[RoR]$ and a risk value around

Table 4.5: Kolmogorov-Smirnov tests between GP (control) and nine ML algorithms (first column) for $\mathbb{E}[RoR]$ and risk. p-values (second column) below the adjusted significance level (third column) appear in boldface to indicate statistical significance at 5% level. The calculation of the adjusted significance level is shown in brackets in the third column.

(a) Expected Rate of Return

Algorithm	p-value	Adj. significance level
SVC	**6.75E-06**	0.0056 (0.05/9)
DT	0.0125	0.0063 (0.05.8)
MLP	0.0227	0.0071 (0.05/7)
GB	0.0302	0.0083 (0.05/6)
PAC	0.0397	0.010 (0.05/5)
SGD	0.1087	0.013 (0.05/4)
ET	0.2557	0.017 (0.05/3)
RF	0.3696	0.025 (0.05/2)
KNN	0.8260	0.05 (0.05/1)

(b) Risk

Algorithm	p-value	Adj. significance level
DT	**1.54E-31**	0.0056 (0.05/9)
PAC	**1.77E-21**	0.0063 (0.05.8)
GB	**4.56E-21**	0.0071 (0.05/7)
MLP	**1.16E-20**	0.0083 (0.05/6)
SGD	**3.29E-11**	0.010 (0.05/5)
SVC	**3.29E-11**	0.013 (0.05/4)
KNN	**1.86E-08**	0.017 (0.05/3)
RF	**1.04E-07**	0.025 (0.05/2)
ET	**6.78E-05**	0.05 (0.05/1)

Table 4.6: GP's average (standard deviation) performance under different stock markets. Best value per financial metric is shown in boldface.

Indexes	**Expected Rate of Return**	**Risk**
DAX	-0.6112% (0.0715)	0.0317 (0.0495)
DJIA	0.3523% (0.0701)	0.0391 (0.0645)
FTSE100	1.5427% (**0.0229**)	0.0350 (0.0369)
NASDAQ	**1.6302%** (0.1200)	0.0823 (0.1187)
NIFTY 50	1.2593% (0.0539)	0.0368 (0.0495)
NIKKEI 225	-1.5516% (0.0634)	**0.0210** (0.0380)
NYSE	-0.4442% (0.0574)	0.0494 (0.0787)
RUSSELL 2000	-0.8550% (0.0984)	0.0476 (0.0654)
S&P500	0.6264% (0.0378)	0.0233 (**0.0308**)
TSEC	0.4271% (0.0232)	0.0290 (0.0373)

0.03, indicating a higher return relative to their risk. The best average $\mathbb{E}[RoR]$ is 1.6302% on the NASDAQ. Risk values range from around 0.03 to 0.05, with NASDAQ being the only exception, having a higher risk of 0.0823. Additionally, the standard deviation values for each index are similar in terms of expected RoR and risk. In general, GP achieved good and stable performance on each index.

In addition, we split the results into 6 countries based on where each market is located. Table 4.7 shows the average result for each country's market. Again we can see good performances in terms of risk across all countries, especially for China, showing that the GP performs stably, regardless of the dataset used. In terms of $\mathbb{E}[RoR]$, results are less uniform, with the lowest $\mathbb{E}[RoR]$ being observed for Japan (-1.5516) and Germany (-0.6112), respectively.

Table 4.7: GP's average performance under different countries

Country	Expected Rate of Return	Risk
US	0.4282%	0.0464
China	0.4271%	0.0290
Germany	-0.6112%	0.0317
Japan	-1.5516%	**0.0210**
UK	**1.5427%**	0.0350
India	1.2593%	0.0368

Table 4.8: Average result for GP on 5 years versus 10 years

GP	Expected Rate of Return	Risk
5 years	-0.5632%	0.0442
10 years	1.0384%	0.0349

In summary, for $\mathbb{E}[RoR]$, GP's performance varies widely across countries and markets from -1.5516% to 1.6302%. For risk, GP's performance is relatively close to each country and market and is very good at around 0.03.

4.4.3 Periods

From time period view, we run GP for two different periods: 5 years (from 2015 to 2020) and 10 years (from 2010 to 2020). Our goal was to investigate whether longer data is beneficial, or if it adds unnecessary noise, given that values from so long ago might contain information that is no longer relevant to the current state of the market [12]. Table 4.8 shows the average result of GP for 5 and 10 years. From Table 4.8, it can be observed that $\mathbb{E}[RoR]$ on 10 years GP is higher than 5 years GP. Risk is at similar levels for both time periods, with the 10 year period risk being slightly lower. To conclude, the information presented in Table 4.8 shows that 10 years' worth of data is more beneficial in terms of expected RoR and risk.

4.4.4 Buy-and-hold strategy

So far, we have evaluated GP's forecasting performance by comparing it with ML algorithms and investigating its performance across different markets and data periods. In this section, we will look at the differences rising between the GP and the popular financial benchmark of buy-and-hold strategy. We would like to reiterate that we use total return (TR) instead of

$\mathbb{E}[RoR]$ and risk as the evaluation metric, as the buy-and-hold strategy has only one trade during the entire trading period, making $\mathbb{E}[RoR]$ and risk metrics unsuitable.

Before proceeding with the comparison, we noticed that, on many occasions, the GP model was deciding not to perform any trades. In fact, of the 220 datasets, the GP traded in only 155 of them. On the other hand, buy-and-hold always performs a trade, given that it buys one amount of stock on the first day of the data and then sells it on the last. To make the comparison between the two algorithms fairer, we used the 155 datasets instead of the 220. In addition, we observed that both the GP and buy-and-hold contained outliers, which could significantly skew the distribution results. To deal with this issue, we removed these outliers by only using those results that were within three SDs of the median. In the end, we removed four outliers from the GP and three outliers from the buy-and-hold.

Our results yielded a TR of around 6.11% for the GP and 14.05% for buy-and-hold. Other ML algorithms were similar to the GP TR performance. This could be explained by two factors. Firstly the data period we have used is predominately during a bull market, especially when we take into account the first and last day of each stock. Thus this puts the buy-and-hold strategy in a very advantageous position. In addition, the GP algorithm was trained by having the Sharpe ratio as its fitness function. However, as we cannot calculate the Sharpe ratio for buy-and-hold[1], we can only compare the TR for GP and buy-and-hold strategy.

4.5 Conclusion

To conclude this chapter, the main contribution is an in-depth comparison of a GP algorithm against different ML algorithms. Experiments took place over 220 datasets derived from 10 international markets. We have shown that GP was able to statistically outperform most algorithms in terms of risk, while it also returned profitable results. This is an important finding because, until now, published works have tended to focus on fewer ML algorithms and/or fewer datasets. Further analysis also showed the differences in terms of international indices and market performance.

[1]There's only two trades, one at the beginning and one at the end, and thus the standard deviation, which is the denominator in the Sharpe ratio formula, cannot be calculated

GP demonstrated a profitable trading strategy with significantly lower risk than for the nine ML algorithms selected. However, the experiments also showed that the SVC significantly outperformed our proposed GP-based algorithm, which motivates us to continue exploring GP's potential using event-based predictions, particularly when focusing on DC. In the next chapter, we will assess the performance of the GP algorithm when applied within the DC framework and compare it against the same GP-based algorithm operating under a physical time framework. Additionally, we propose an exploration of a combination approach that integrates both physical time and event-based prediction methodologies. This investigation aims to ascertain whether such a combined method can yield an enhanced performance.

Chapter 5

GP application to an event-based framework

5.1 Motivation

In the last chapter, we compared a GP-based algorithm under the physical time paradigm alongside several other ML algorithms. The result indicated that GP is quite competitive when compared with most other algorithms, especially when considering a panel of factors. However, the GP-based algorithm was significantly outperformed by the SVC algorithm.

In this chapter, we explore the potential of directional changes (DC) as a complementary technique to physical time. Our goal is to achieve performance improvements that outperform the nine ML algorithms using DC-based indicators. As discussed in Chapter 3, DC has been applied successfully in the domain of algorithmic trading, although predominantly in the Forex market, leaving its application in the stock market largely unexplored. Furthermore, there is no published work, to the best of our knowledge, that utilises a large number of DC-based indicators.

Given GP's ability to effectively combine diverse indicators and create profitable trading strategies [117, 144, 145, 2], this chapter will present a novel GP algorithm that utilizes 28 DC-based indicators. We call this GP as GP-DC. We aim to demonstrate that the GP can create novel trading strategies that yield competitive trading performance. For completeness, we

also propose another GP algorithm, namely GP-DC-PT, which will use both 28 DC indicators and 28 physical time (technical analysis) indicators. These two novel GP-based algorithms are benchmarked against the GP algorithm presented in Chapter 4. To distinguish the two, we will refer to this algorithm as GP-PT. Experiments will be conducted on the same 220 datasets as used in the previous chapter.

The rest of this chapter is organised as follows. The proposed methodology is provided in Section 5.2, while Section 5.3 presents the benchmarks and the parameter tuning process. Then, in Section 5.4 we discuss the results of our experiments and conclude this chapter in Section 5.5.

5.2 Methodology

As the three GP algorithms explored within this chapter share several characteristics with the GP in Chapter 4, we will present only those components of the algorithms that differ from those of the previous chapter. Hence, Section 5.2.1 will introduce indicators that will be used by the algorithms, and Section 5.2.2 will present the representation of the GP individuals under GP-DC, GP-PT, and GP-DC-PT.

5.2.1 Directional changes and technical indicators

We use 28 specific DC indicators, which have been previously introduced and discussed in [5]. These DC indicators are conceptually similar to the technical indicators used in traditional time-based technical analysis, as detailed in [152]. Again, we employ the ERC that generates a random number distributed uniformly between -1 and 1. Therefore, the DC indicators have been appropriately normalised.

Table 5.1 provides a comprehensive listing of these 28 DC indicators. Notably, among these indicators, 10 involve calculations over specific time intervals, such as the total count of DC events denoted as N_{DC}, which can be computed over varying time windows, including 10, 20, 30, 40, or 50 days. This brings the total count of indicators to 28. For indicators that do not depend on a specific time window, such as OSV, TMV, T_{DC}, and R_{DC}, Table 5.1

Table 5.1: DC indicators; see also [5] and [6]

Indicator	Periods (days)
Total price movements value at extreme points (TMV) [Equation 2.12]	N/A
Overshoot Values at Extreme Points (OSV) [Equation 2.13]	N/A
Average OSV [Equation 2.14]	3, 5, 10
Time-adjusted return of DC (R_{DC}) [Equation 2.15]	N/A
Average R_{DC} [Equation 2.16]	3, 5, 10
Time for completion of a trend (T_{DC}) [Definition in 2.1.4]	N/A
Average T_{DC} [Equation 2.17]	3, 5, 10
Number of directional change events (N_{DC}) [Definition in 2.1.4]	10, 20, 30, 40, 50
Time independent Coastline (C_{DC}) [Equation 2.18]	10, 20, 30, 40, 50
Up and down trends asymmetry in time intervals (A_T) [Equation 2.19]	10, 20, 30, 40, 50

designates them as 'N/A.'

In addition to the above 28 DC-based indicators, our study also incorporates 28 PT indicators derived from technical analysis to complement our analytical framework. It's important to note that, in the preceding chapter, a more extensive set of 146 PT technical indicators was utilised. However as the GP-DC-PT algorithm will be using both DC and physical time indicators in its terminal set, it was important to have an equal number of indicators to avoid the GP being biased towards PT indicators. Hence the number of PT indicators used by the GP-DC-PT is restricted to 28, as per the DC indicators. In addition, we will also use the same 28 PT indicators for GP-PT to ensure a fair comparison among the GP algorithms. This choice of PT indicators is informed by the work of [7], which highlights the most common indicators within the domain of financial forecasting. Some of these indicators (MA, CCI, RSI, and William's %R) are considered for periods of 10, 20, 30, 40, and 50 days, and a couple of them (ATR and EMA) are considered for periods of 3, 5, and 10 days, while finally, OBV and PSAR do not involve periods. The above periods are commonly used values in the literature [153]. Table 5.2 summarises these indicators.

The proposed GP-DC algorithm uses only DC indicators in its terminal set and thus only the indicators presented in Table 5.1. On the other hand, GP-PT uses only physical time technical analysis indicators in its terminals and therefore the indicators presented in Table 5.2. Lastly, the proposed GP-DC-PT algorithm uses both DC and technical analysis indicators in its terminal set, i.e. the indicators from both Tables 5.1 and 5.2.

Table 5.2: Physical time (technical analysis) indicators; see also [7]

Indicator	Periods (days)
Moving average (MA) [Equation 2.2]	10, 20, 30, 40, 50
Commodity Channel Index (CCI) [Equation 2.3 to 2.5]	10, 20, 30, 40, 50
Relative Strength Index (RSI) [Equation 2.1]	10, 20, 30, 40, 50
William's %R [Equation 2.6]	10, 20, 30, 40, 50
Average True Range (ATR) [Equation 2.7 and Equation 2.8]	3, 5, 10
Exponential Moving Average (EMA) [Equation 2.9]	3, 5, 10
On Balance Volume (OBV) [Equation 2.10]	N/A
Parabolic Stop and Reverse (PSAR) [Equation 2.11]	N/A

5.2.2 Model representation

Figure 5.1 presents illustrative tree structures corresponding to the GP-PT, GP-DC, and GP-DC-PT algorithms. To ensure comparability, identical tree structures are maintained across the examples, with variations confined to the terminal sets. As depicted in Figure 5.1, the GP-PT and GP-DC algorithms generate trees with all PT and DC indicators. Conversely, the GP-DC-PT algorithm possesses the capability to construct trees incorporating both PT and DC indicators. It is worth noting that GP-DC-PT could also produce a GP tree with all PT or DC indicators.

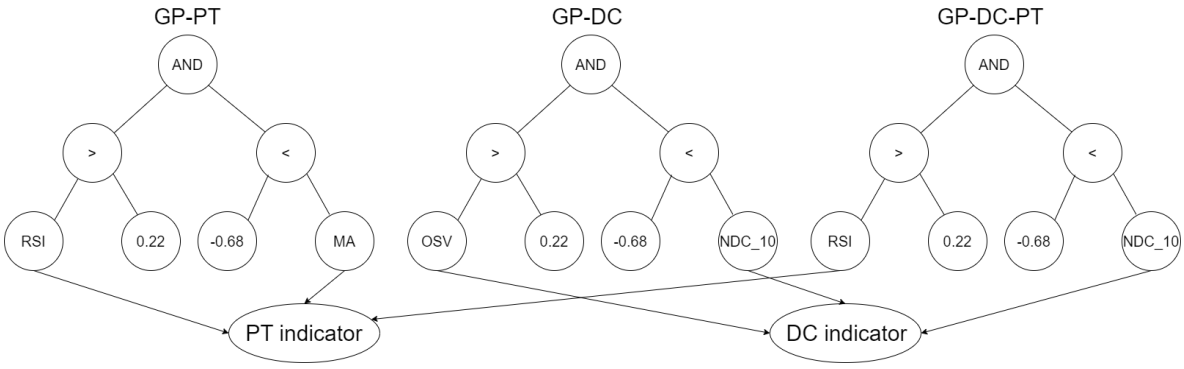


Figure 5.1: An example tree for GP-PT, GP-DC, and GP-DC-PT.

5.2.3 Remainder of the GP configuration

As discussed earlier, all three GPs follow the configurations of the GP introduced in Chapter 4. Hence, the Function set is the same across all of them, i.e., AND, OR, less than ($<$), and

greater than ($>$). Similarly, they all follow the same trading strategy as previously introduced in Section 4.2.2. The Sharpe ratio is again the fitness function employed across all GP algorithms. In addition, the same genetic operators are applied as presented in Section 4.2.1.4.

5.3 Experimental set-up

In this section, because we use the same data introduced in Section 4.3.1, we only introduce the benchmarks of the experimental comparison and parameter tuning.

5.3.1 Benchmarks

The two novel GP algorithms, namely GP-DC and GP-DC-PT, are benchmarked against GP-PT, which only uses physical time. This allows us to understand the added value of using GP algorithms that include DC-based indicators, rather than those that incorporate physical time.

Our second benchmark is the nine ML algorithms presented in Section 4.3, as one of the main goals of this chapter is to enhance the GP-based algorithm to outperform these nine algorithms. In addition, we also benchmark the proposed algorithm against trading strategies derived from physical time technical analysis indicators. Such indicators are very common in the financial literature and the trading industry, so we consider it important to compare our proposed algorithms' performance against such indicators. We run trading experiments for the following indicators: Moving Average Convergence/Divergence (MACD), On Balance Volume (OBV), and Momentum (MTM). For MACD and OBV, a buy signal is produced when their long-term moving average (50 days) has a higher value than their short-term moving average (10 days); and vice versa for a sell action. For MTM, a buy signal is generated when the current price is higher than the price of 10 days ago, and vice versa for the sell action. All of the above values (10 and 50 days for MACD and OBV; 10 days for MTM) were selected through a grid search.

Finally, the final benchmark is the buy-and-hold strategy. As introduced in Section 4.3.2,

the buy-and-hold strategy is another traditional approach that could be used in finance forecasting. The motivation for including this trading strategy is because it represents a different type of trading preference, namely a passive one, where traders buy a stock and hold it for a lengthy period. Given that the three GPs used in this chapter employ different indicators to the GP in Chapter 4, it is important to evaluate their performance against buy-and-hold.

5.3.2 Parameter tuning

In Section 4.3.3, we executed a grid search to identify the optimal GP parameters and two key trading strategy parameters for the proposed GP-based algorithms. This chapter extends this optimisation process to include a novel parameter derived from the DC framework, designated θ. It controls the magnitude of the DC event under the DC framework. In other words, it decides what event could be viewed as significant, which, traditionally, should be specified by the users. Similarly to the n and r parameters, we also conduct a grid search for θ across all 220 datasets. The advantage of this is that each dataset will be using a tailored value of θ and, as a result, a more appropriate generated event-based series. Hence, GP-DC and GP-DC-PT will perform a grid search for all three parameters n, r, and θ, while GP-PT, which does not use directional changes, will perform a grid search for n and r only. The range of values for the grid search tuning for n, r, and θ, are as follows:

- n (prediction horizon): 1, 5, 15
- r (price movement percentage): 1%, 5%, 10%, 20%
- θ (DC threshold): 0.001, 0.002, 0.005, 0.01, 0.02

5.4 Results and analysis

The goal of our experiments was to explore the potential of DC as a complementary technique to physical time. To achieve this goal, we evaluated GP-DC using GP-PT as a benchmark, and we further studied GP-DC-PT to evaluate whether the two indicator sets can complement each other.

In Section 5.4.1.1, we compare GP-DC and GP-DC-PT with GP-PT over distinct period lengths of 5 and 10 years. In this section, we provide comprehensive results and insights across all four metrics: total return (TR), expected rate of return ($\mathbb{E}[RoR]$), risk, and the Sharpe ratio. Once the above comparisons are complete, we identify the best GP algorithm and compare it against the benchmarks of technical indicators (Section 5.4.3) and buy-and-hold (Section 5.4.4).

5.4.1 Comparison of the GP-based algorithms

5.4.1.1 Periods

Recall that our datasets were created over two periods, i.e. for 5 and 10 years, respectively. In this section, we will consider the influence of the period on the performance of the three GP-based algorithms. In addition to TR, $\mathbb{E}[RoR]$, and risk, as evaluation metrics, we now also consider the portfolio Sharpe ratio for GP-DC, GP-PT, and GP-DC-PT. The portfolio Sharpe ratio assumes an equal distribution of the 110 stocks (i.e., each stock has the same weight in the portfolio). Since it is calculated across the entire portfolio of all stocks, there are no summary statistics to calculate, just a single value, which is presented in the table.

As we can observe in Table 5.3, our proposed algorithms perform very well over the 5-year period, particularly in terms of TR and $\mathbb{E}[RoR]$. GP-DC has the highest average (13.89%) and median (9.63%) values for both metrics, while GP-DC-PT comes second. With regards to risk, GP-DC-PT again yields the lowest average (0.06) and median (0.05) values, while GP-PT comes second and GP-DC marginally third. When looking at the 10-year data, we find that GP-PT has the highest average TR (21.38%), which is probably affected by the outliers (e.g. maximum value 571.21%)[1]. GP-DC has the highest median TR (10.96%). In terms of $\mathbb{E}[RoR]$, GP-PT has the best average (2.47%) and median (1.00%) values. Furthermore, in terms of risk, GP-DC-PT is again showing strong performance, as it has the lowest values, with GP-DC coming second.

[1]Such outliers can be explained by a combination of a heavily bull market and a very passive trading strategy. For example, a particular GP model might recommend a buy action at the beginning of the 2-year test set period (for the 10-year dataset), and then sell towards the end of the test set period. Assuming that the respective dataset's prices were in a strong upward trend, such an action can lead to a very profitable trade.

Table 5.3: Summary statistics of the GP-based algorithms on 5 and 10-year periods. We use boldface for the best values for each metric.

Period	Measure	Algorithm		
		Total return		
		GP-DC	GP-PT	GP-DC-PT
5 year	Average	**13.89**%	2.82%	10.23%
	Median	**9.63**%	3.83%	9.02%
	Standard deviation	0.28	0.21	**0.18**
	Maximum	**187.98**%	130.44%	130.98%
	Minimum	**-33.08**%	-57.50%	-40.45%
10 year	Average	15.91%	**21.38**%	14.29%
	Median	**10.96**%	10.13%	10.85%
	Standard deviation	**0.37**	0.77	0.38
	Maximum	147.52%	**571.21**%	186.61%
	Minimum	-149.59%	**-53.59**%	-229.56%
		Expected Rate of Return		
		GP-DC	GP-PT	GP-DC-PT
5 year	Average	**1.58**%	0.58%	1.13%
	Median	**1.37**%	0.61%	1.17%
	Standard deviation	**0.02**	0.03	**0.02**
	Maximum	**10.07**%	9.08%	7.74%
	Minimum	-7.55%	-10.93%	**-3.82**%
10 year	Average	1.28%	**2.47**%	0.91%
	Median	0.89%	**1.00**%	0.77%
	Standard deviation	0.03	0.08	**0.01**
	Maximum	27.00%	**46.06**%	6.00%
	Minimum	**-2.83**%	-22.31%	-3.33%
		Risk		
		GP-DC	GP-PT	GP-DC-PT
5 year	Average	0.10	0.09	**0.06**
	Median	0.08	0.07	**0.05**
	Standard deviation	0.08	0.07	**0.04**
	Maximum	0.48	0.45	**0.22**
	Minimum	0.01	**0.00**	0.01
10 year	Average	0.08	0.10	**0.06**
	Median	0.07	0.07	**0.05**
	Standard deviation	0.06	0.10	**0.04**
	Maximum	0.39	0.53	**0.24**
	Minimum	0.01	0.01	0.01

Table 5.4: Sharpe ratio of the GP-based algorithms on 5 and 10-year periods. We use boldface for the best values for each period.

Period	GP-DC	GP-PT	GP-DC-PT
5 year	0.5451	0.1867	**0.7214**
10 year	0.4266	0.3100	**0.7081**

Finally, for a comprehensive evaluation of each algorithm's risk-adjusted return performance, we establish portfolios for each GP variant (GP-DC, GP-PT, GP-DC-PT) within the 5 and 10-year time frames. This results in six distinct portfolios, each comprising 110 datasets with an equal allocation weight per dataset.[2] This methodology allows us to gain a holistic perspective on the performance of each algorithm across different periods. Thus, in Table 5.4, we present the Sharpe ratio performance of each GP's portfolio. It is clear that the ranking remained consistent for both the 5-year and 10-year periods, with GP-DC-PT achieving the top result, followed by GP-DC in second place, and GP-PT trailing in third. It reveals the consistent profit-generating capability of GP-DC-PT.

We use the non-parametric KS test, together with the Holm-Bonferroni correction, to evaluate the above results. Table 5.5 and Table 5.6 present the p-values of the KS test between the GP-DC algorithm (control algorithm) and each of the GP-PT and GP-DC-PT algorithms in terms of TR, $\mathbb{E}[RoR]$, and risk; p-values indicating statistical significance appear in boldface.

We remark that the distribution of GP-DC and GP-PT algorithms is statistically and significantly different in terms of TR and $\mathbb{E}[RoR]$ in the 5-year datasets. Combined with Table 5.3, we can conclude that GP-DC outperforms, in a statistically significant manner, the GP-PT algorithm in terms of TR and $\mathbb{E}[RoR]$ for these datasets. Meanwhile, the difference between GP-DC and GP-DC-PT is also significant in terms of 5-year and 10-year risk. This further shows a considerable improvement in risk for the GP-DC-PT algorithm. For the rest of the comparisons, the differences in performance are not statistically significant.

In conclusion, the GP-DC algorithm generally performs better than the GP-PT algorithm in the short term (5 years). In contrast, the GP-PT algorithm generally outperforms the GP-DC algorithm over the long term (10 years), whereas the GP-DC and GP-PT algorithms exhibit statistically significant differences in TR and $\mathbb{E}[RoR]$ in 5 years (both TR and $\mathbb{E}[RoR]$ are higher for GP-DC). Moreover, the GP-DC-PT algorithm significantly reduces the risk with a similar distribution of TR to the GP-DC algorithm in both 5- and 10-year datasets. In addition, the higher Sharpe ratio of the GP-DC-PT algorithm reflects stability. Apart from providing

[2]It's worth noting that this thesis primarily centres on algorithmic trading, and the optimization of portfolio weights falls beyond its current scope. Therefore, for simplicity, equal weights have been assigned. Future research will delve into the optimization of portfolio allocations.

Table 5.5: Kolmogorov-Smirnov tests between GP-DC (control) with GP-PT and GP-DC-PT for TR, $\mathbb{E}[RoR]$, and risk on 5 years. p-values (second column) below the adjusted significance level (third column) appear in boldface to indicate statistical significance at a 5% level. The calculation of the adjusted significance level is shown in brackets in the third column.

(a) TR

Algorithm	p-value	Adj. significance level
GP-PT	**0.0034**	0.025 (0.05/2)
GP-DC-PT	0.2395	0.05 (0.05/1)

(b) Expected Rate of Return

Algorithm	p-value	Adj. significance level
GP-DC-PT	**0.0055**	0.025 (0.05/2)
GP-PT	**0.0137**	0.05 (0.05/1)

(c) Risk

Algorithm	p-value	Adj. significance level
GP-DC-PT	**6.9784e-05**	0.025 (0.05/2)
GP-PT	0.5060	0.05 (0.05/1)

Table 5.6: Kolmogorov-Smirnov tests between GP-DC (control) with GP-PT and GP-DC-PT for TR, $\mathbb{E}[RoR]$, and risk on 10 years. p-values (second column) below the adjusted significance level (third column) appear in boldface to indicate statistical significance at 5% level. The calculation of the adjusted significance level is shown in brackets in the third column.

(a) TR

Algorithm	p-value	Adj. significance level
GP-PT	0.5060	0.025 (0.05/2)
GP-DC-PT	0.6186	0.05 (0.05/1)

(b) Expected Rate of Return

Algorithm	p-value	Adj. significance level
GP-DC-PT	0.2395	0.025 (0.05/2)
GP-PT	0.5060	0.05 (0.05/1)

(c) Risk

Algorithm	p-value	Adj. significance level
GP-DC-PT	**7.2247e-04**	0.025 (0.05/2)
GP-PT	0.2395	0.05 (0.05/1)

further evidence that DC-based algorithms could complement physical time methods, this demonstrates the advantage of the GP-DC-PT algorithm over the GP-DC algorithm.

5.4.1.2 Summary

By comparing the three GP-based algorithms' performance over different periods, we conclude that both DC-based algorithms outperformed GP-PT. This remark fulfils the objective of this chapter, i.e. to demonstrate that DC-based trading strategies constitute a promising alternative to traditional physical time-based approaches. Furthermore, the GP-DC-PT algorithm has a significantly lower risk than the GP-DC algorithm. Meanwhile, although the GP-DC algorithm achieved better TR in many cases, the difference between GP-DC and GP-DC-PT algorithms is not statistically significant. Finally, when analysing the Sharpe ratio results, it

becomes evident that GP-DC-PT exhibits significantly higher values compared to GP-DC, with approximately a 33% improvement for the 5-year period and a remarkable 66% improvement for the 10-year period (refer to Table 5.4). The Sharpe ratio, being a comprehensive metric that incorporates both returns and risk, carries substantial significance, mirroring real-world behaviours of financial analysts and traders. Therefore, we emphasise the importance of placing greater weight on the insights derived from this metric. Consequently, we assert that, when it comes to striking a balance between return and risk, GP-DC-PT emerges as the preferred choice among the three GP-based algorithms. As a result, the subsequent sections will involve benchmarking the GP-DC-PT algorithm against technical indicators and a traditional buy-and-hold trading strategy.

5.4.2 Comparison of GP-DC-PT algorithm to the nine ML algorithms

We now compare GP-DC-PT with the nine ML algorithms tested previously in Chapter 4, as the GP algorithm was not able to outperform some of them. Note that in order to make a fair comparison to GP-DC-PT, they all use the same 28 physical time indicators (rather than the 146 indicators used in the previous chapter). The summary statistics of the comparison are given in Table 5.7 and Table 5.8. Note that GP-DC-PT has the best average and median TR in 5 years, three times higher than the rest of the algorithms. In terms of the expected RoR, the GP-DC-PT ranks first on all metrics except for the maximum value, indicating a strong overall performance. In terms of risk, all algorithms show similarly low risk values, with SVC having the lowest average and median. Importantly, GP-DC-PT achieved the best Sharpe ratio of 0.72, which is also three times higher than achieved with the other algorithms.

Over the 10-year period, the advantage of GP-DC-PT persists in terms of TR, ranking second in average value, just 0.06% below the best performer, while showing a significant improvement in median value. While SGD outperforms in terms of both the average and median expected rate of return, its median value is only 0.10% higher than GP-DC-PT's (0.87% vs 0.77%). In contrast, although GP-DC-PT has a slightly lower average and median $\mathbb{E}[RoR]$ compared to SGD, it produces more stable profits as the standard deviation is five times lower than SGD's (0.01 vs 0.05). Additionally, the risk level of GP-DC-PT remains consistent with

that observed in the 5-year period. Lastly, the better performance of GP-DC-PT is further highlighted by its top Sharpe ratio of 0.71.

To compare the GP-DC-PT with nine ML algorithms, we use the Friedman test to measure the average rank among them. As we can observe from Table 5.9 and Table 5.13, the GP-DC-PT algorithm ranks first on four out of six comparisons. More specifically, GP-DC-PT statistically and significantly outperforms all other ML algorithms in terms of total return under the 5-year period. In terms of $\mathbb{E}[RoR]$, it again ranks first and statistically and significantly outperforms PAC, SVC, RF, GB, and KNN. In terms of risk, PAC ranks first and statistically and significantly outperforms GP-DC-PT at the 5% significance level. Results are very similar for the 10-year period. However, it is worth noting that GP-DC-PT's risk is improved and ranks third and is not statistically outperformed by the control algorithm (RF).

Table 5.7: Summary statistics of the GP-DC-PT algorithm and other nine machine learnings on a 5-year period. Best value per row is denoted in boldface.

Total return										
	GP-DC-PT	DT	ET	GB	KNN	MLP	PAC	RF	SGD	SVC
Average	**10.24**%	3.17%	3.08%	1.91%	2.42%	2.61%	1.89%	3.25%	2.87%	2.40%
Median	**9.03**%	2.19%	1.85%	1.23%	0.00%	2.03%	2.18%	1.42%	2.30%	0.00%
StDev	0.18	0.19	0.14	0.16	0.16	0.14	**0.11**	0.15	0.12	0.12
Max	130.98%	**78.52**%	43.82%	65.94%	69.84%	43.33%	31.75%	58.18%	48.82%	59.74%
Min	-40.45%	-85.46%	-33.21%	-62.78%	-60.19%	-41.54%	-32.69%	**-29.85**%	-37.29%	-33.63%
Expected Rate of Return										
	GP-DC-PT	DT	ET	GB	KNN	MLP	PAC	RF	SGD	SVC
Average	**1.14**%	0.67%	0.70%	0.26%	0.50%	0.48%	0.57%	0.51%	0.59%	0.48%
Median	**1.17**%	0.44%	0.32%	0.35%	0.00%	0.31%	0.59%	0.25%	0.45%	0.00%
StDev	**0.02**	0.03	0.04	0.04	**0.02**	0.03	0.03	0.03	**0.02**	**0.02**
Max	7.74%	16.52%	17.50%	**16.80**%	10.64%	8.38%	8.45%	8.58%	7.62%	7.24%
Min	**-3.82**%	-11.30%	-16.77%	-17.44%	-5.76%	-10.13%	-7.59%	-7.18%	-8.31%	-6.69%
Risk										
	GP-DC-PT	DT	ET	GB	KNN	MLP	PAC	RF	SGD	SVC
Average	0.07	0.07	0.06	0.06	0.06	0.06	0.06	0.06	0.06	**0.05**
Median	0.06	0.05	0.04	0.04	0.05	0.05	0.05	0.05	0.05	**0.04**
StDev	**0.04**	0.05	0.06	0.06	0.05	0.05	**0.04**	0.05	0.05	**0.04**
Max	**0.22**	0.39	0.32	0.34	0.34	0.28	0.27	0.27	0.31	0.24
Min	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Sharpe ratio										
	GP-DC-PT	DT	ET	GB	KNN	MLP	PAC	RF	SGD	SVC
	0.72	0.20	0.19	0.07	0.23	0.17	0.22	0.17	0.25	0.21

Table 5.8: Summary statistics of the GP-DC-PT algorithm and other nine machine learnings on a 10-year period. Best value per row is denoted in boldface.

Total return										
	GP-DC-PT	DT	ET	GB	KNN	MLP	PAC	RF	SGD	SVC
Average	14.29%	4.57%	6.72%	10.53%	9.78%	9.51%	9.29%	8.78%	**14.35**%	11.78%
Median	**10.85**%	0.00%	1.44%	1.52%	0.72%	1.35%	3.29%	1.65%	4.32%	0.00%
StDev	0.38	**0.25**	0.30	0.37	0.40	0.36	0.30	0.31	0.37	0.40
Max	186.62%	125.52%	128.85%	263.53%	**323.73**%	225.97%	232.80%	132.87%	231.65%	259.37%
Min	-229.56%	-51.58%	-114.26%	**-43.88**%	-95.37%	-50.44%	-61.89%	-45.46%	-72.21%	-50.13%
Expected Rate of Return										
	GP-DC-PT	DT	ET	GB	KNN	MLP	PAC	RF	SGD	SVC
Average	0.91%	0.42%	0.78%	0.88%	0.75%	0.86%	1.05%	1.14%	**1.70**%	0.95%
Median	0.77%	0.05%	0.31%	0.28%	0.17%	0.29%	0.56%	0.37%	**0.87**%	0.00%
StDev	**0.01**	0.02	0.03	0.03	0.02	0.02	0.03	0.05	0.05	0.03
Max	6.00%	7.45%	14.59%	21.45%	17.48%	14.26%	15.20%	**32.56**%	25.55%	19.15%
Min	**-3.34**%	-5.58%	-9.68%	-12.02%	-6.10%	-7.13%	-11.49%	-11.84%	-7.39%	-4.92%
Risk										
	GP-DC-PT	DT	ET	GB	KNN	MLP	PAC	RF	SGD	SVC
Average	0.06	0.06	0.07	0.07	**0.05**	0.06	0.07	0.07	0.07	0.07
Median	0.05	0.05	0.05	0.05	**0.04**	**0.04**	0.05	0.05	0.05	0.05
StDev	**0.04**	0.07	0.09	0.07	0.05	0.05	0.06	0.08	0.06	0.09
Max	0.38	0.51	0.58	0.47	0.38	0.31	0.38	0.55	**0.30**	0.68
Min	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Sharpe ratio										
	GP-DC-PT	DT	ET	GB	KNN	MLP	PAC	RF	SGD	SVC
	0.71	0.22	0.23	0.25	0.29	0.34	0.34	0.21	0.36	0.32

Table 5.9: Friedman with Bonferroni's post-hoc test between GP-DC-PT algorithms and ML algorithms on a 5-year period. Statistically significant differences at a 5% level are shown in boldface.

Table 5.10: Total return

Algorithm	Average rank	p_{Bonf}
GP-DC-PT(c)	4.063636	-
ET	5.281818	**5.48E-3**
SVC	5.372727	**2.92E-3**
MLP	5.445455	**2.41E-3**
SGD	5.445455	**2.41E-3**
DT	5.454545	**2.41E-3**
RF	5.672727	**3.86E-4**
PAC	5.754545	**2.05E-4**
KNN	5.854545	**1.12E-4**
GB	5.5.918182	**1.12E-4**

Table 5.11: Expected Rate of Return

Algorithm	Average rank	p_{Bonf}
GP-DC-PT(c)	4.381818	-
ET	5.190909	0.14
SGD	5.218182	0.14
DT	5.354545	0.08
MLP	5.381818	0.07
PAC	5.490909	**0.04**
SVC	5.645455	**0.01**
RF	5.790909	**4.87E-3**
GB	5.881818	**3.36E-3**
KNN	6.000000	**1.80E-3**

Table 5.12: Risk

Algorithm	Average rank	p_{Bonf}
PAC(c)	4.881818	-
MLP	4.936364	0.65
RF	4.990909	0.61
ET	5.036364	0.61
SGD	5.090909	0.59
GB	5.190909	0.43
KNN	5.663636	0.08
DT	5.763636	0.07
GP-DC-PT	6.263636	**3.80E-3**
SVC	6.936364	**1.20E-5**

From the above, we can conclude that this chapter's proposed GP-DC-PT is a competitive algorithm and is able to do much better when compared to other ML algorithms in terms of total return and expected rate of return. When it comes to risk, ML algorithms perform better. However, we should keep in mind that qualitatively the differences in performance are not large, e.g. in Table 5.7 we saw that the differences in average and median risk values between GP-DC-PT and the best performing algorithm (SVC) were only 0.02%. In terms of the 10-year period (Table 5.8), the differences were even smaller. In addition to the above, we should always keep in mind that traders don't consider metrics in isolation as we discussed

Table 5.13: Friedman with Bonferroni's post-hoc test between GP-DC-PT algorithms and ML algorithms on a 10-year period. Statistically significant differences at a 5% level are shown in boldface.

Table 5.14: Total return

Algorithm	Average rank	p_{Bonf}
GP-DC-PT(c)	4.227273	-
SGD	4.900000	0.20
SVC	5.245455	**1.93E-2**
KNN	5.309091	**1.93E-2**
MLP	5.318182	**1.74E-2**
PAC	5.445455	**9.20E-3**
GB	5.518182	**7.77E-3**
RF	5.700000	**1.55E-3**
ET	5.727273	**1.55E-3**
DT	5.836364	**7.83E-4**

Table 5.15: Expected Rate of Return

Algorithm	Average rank	p_{Bonf}
GP-DC-PT(c)	4.60000	-
SGD	4.618182	0.86
PAC	5.272727	0.18
MLP	5.300000	0.15
SVC	5.418182	0.10
KNN	5.436364	0.10
GB	5.554545	0.08
RF	5.618182	0.06
ET	5.718182	0.05
DT	5.827273	**0.03**

Table 5.16: Risk

Algorithm	Average rank	p_{Bonf}
RF(c)	4.890909	-
KNN	5.027273	0.77
GP-DC-PT	5.272727	0.76
PAC	5.336364	0.69
DT	5.381818	0.54
GB	5.381818	0.57
SGD	5.381818	0.68
MLP	5.436364	0.53
ET	5.445455	0.53
SVC	6.809091	**3.30E-5**

in Section 2.3. So while return and risk offer valuable insights, an aggregate metric like the Sharpe ratio is preferred when comparing trading strategies. Hence, the fact that GP-DC-PT has a significantly higher Sharpe ratio than all other ML algorithms indicates that it offers a more favourable risk-adjusted return, making it a stronger overall trading strategy.

5.4.3 Comparison of GP-DC-PT algorithm to other technical indicators

To compare the effectiveness of our approach to traditional technical analysis, we benchmark the GP-DC-PT algorithm against three commonly used indicators: Moving Average Convergence Divergence (MACD), On Balance Volume (OBV), and Momentum (MTM), also used in [154]. The outcome of this comparison can be seen in Table 5.17 and Table 5.18, where a value of 0 means that the respective algorithm did not perform any trades. Note that GP-DC-PT exhibits a very high median TR on both 5- and 10-year data, together with a best average TR on 5-year data and a second-best average TR on 10-year data. In terms of $\mathbb{E}[RoR]$, GP-DC-PT has the second-best average value and the best median on both 5- and 10-year data. With respect to risk, GP-DC-PT has the lowest risk in most cases, except for the median risk derived from the 5-year data. In addition, we calculate the Sharpe ratio of the portfolio, where each of the 110 datasets is given equal weight, for 5- and 10-year datasets separately. The previous findings are also fully reflected in the Sharpe ratio, where GP-DC-PT demonstrated a clear advantage, both in the 5-year and 10-year datasets, relative to the other three indicators.

Table 5.19 and Table 5.20 present the KS test that confirms the above results. As we can observe, by considering the Holm-Bonferroni correction for the three comparisons, all p values below the significance level reject the null hypothesis at the 5% significance level and indicate that the differences are statistically significant in terms of TR, $\mathbb{E}[RoR]$, and risk for both 5- and 10-year periods.

From the discussion above, it is reasonable to conclude that our proposed algorithm, GP-DC-PT, has a significant advantage over more popular technical indicators. For some specific stocks, technical indicators perform well, with even better TR and $\mathbb{E}[RoR]$ than the GP-DC-PT algorithm. On both 5- and 10-year periods, however, the GP-DC-PT algorithm significantly outperforms the other three technical indicators at the 5% significance level.

Table 5.17: Summary statistics of the GP-DC-PT algorithm and technical analysis on a 5-year period. Best value per row is denoted in boldface.

	Total return			
	GP-DC-PT	MACD	OBV	MTM
Average	**10.24**%	7.80%	2.93%	6.72%
Median	**9.03**%	0.01%	0.01%	0.48%
StDev	**0.18**	0.39	0.31	0.48
Max	130.98%	191.07%	152.31%	**252.25**%
Min	**-40.45**%	-97.24%	-98.39%	-98.49%
	Expected Rate of Return			
	GP-DC-PT	MACD	OBV	MTM
Average	1.14%	**1.31**%	-0.03%	0.32%
Median	**1.17**%	0.00%	0.00%	0.02%
StDev	**0.02**	0.07	0.05	**0.02**
Max	7.74%	**31.84**%	21.52%	10.98%
Min	-3.82%	-24.31%	-24.60%	**-2.59**%
	Risk			
	GP-DC-PT	MACD	OBV	MTM
Average	**0.07**	0.09	**0.07**	0.12
Median	0.06	**0.05**	**0.05**	0.08
StDev	**0.04**	0.15	0.07	0.15
Max	**0.22**	1.05	0.35	0.95
Min	0.01	0.01	0.01	0.02
	Sharpe ratio			
	GP-DC-PT	MACD	OBV	MTM
	0.72	0.19	-0.01	0.17

5.4.4 Buy-and-hold strategy

We now compare GP-DC-PT with the state-of-the-art baseline approach, namely the buy-and-hold strategy. Recall that the buy-and-hold strategy only makes one trade during the whole period, during which we buy one unit of stock on the first day of trading, and sell it on the last day. Hence, the risk and $\mathbb{E}[RoR]$ are unsuited for such a comparison.

As we can observe from Table 5.21 and Table 5.22, the GP-DC-PT algorithm has the better median TR on the 5-year period. In contrast, the buy-and-hold strategy performs better on the rest of the average and median TR. It is worth noting that the buy-and-hold strategy achieves the highest maxima (1753.05%), indicating that extreme values cause its high average effect. This is because some of the datasets were derived from a predominately bull

Table 5.18: Summary statistics of the GP-DC-PT algorithms and technical indicators over a 10-year period. Best value per row is denoted in boldface.

	Total return			
	GP-DC-PT	MACD	OBV	MTM
Average	14.30%	13.99%	2.63%	**18.43**%
Median	**10.85**%	2.22%	1.73%	7.44%
StDev	**0.38**	0.58	0.43	0.67
Max	186.62%	360.11%	148.28%	**514.56**%
Min	-229.56%	**-90.38**%	-177.72%	-112.67%
	Expected Rate of Return			
	GP-DC-PT	MACD	OBV	MTM
Average	0.91%	**1.71**%	0.77%	0.32%
Median	**0.77**%	0.14%	0.18%	0.11%
StDev	**0.01**	0.06	0.05	**0.01**
Max	6.00%	36.01%	**35.65**%	6.95%
Min	-3.34%	-9.04%	-11.11%	**-1.48**%
	Risk			
	GP-DC-PT	MACD	OBV	MTM
Average	**0.06**	0.12	0.12	0.15
Median	**0.05**	0.07	0.09	0.09
StDev	**0.04**	0.22	0.12	0.20
Max	**0.38**	1.58	0.76	1.82
Min	0.01	0.01	0.01	0.03
	Sharpe ratio			
	GP-DC-PT	MACD	OBV	MTM
	0.71	0.26	0.15	0.29

market (i.e., there was a strong upward trend in the price series). Therefore, the last price can differ significantly from the first (especially for the 10-year datasets, where the test set is 2 years long). For example, the above return of 1753.05% comes from NASDAQ's Plug Power Inc. which, for the period from 2018-2020 saw its daily price increase sharply due to a combination of factors such as the increasing interest in clean energy, significant partnerships and deals, and strong financial performance. As a result, such a high TR result can significantly affect the mean values, which are sensitive to outliers. It could be told by the better standard deviation of GP-DC-PT. To deal with this issue, we also analysed the results by only using all results within two SDs of the mean. Therefore, the average (standard deviation) TR of the buy-and-hold strategy becomes 3.21% (0.18) and 6.98% (0.19) on the 5- and 10-year period,

Table 5.19: Kolmogorov-Smirnov tests between GP-DC-PT (control) and three technical indicators for TR, $\mathbb{E}[RoR]$, and risk on a 5-year period. p-values (second column) below the adjusted significance level (third column) appear in boldface to indicate statistical significance at the 5% level. The calculation of the adjusted significance level is shown in brackets in the third column.

(a) TR

Algorithm	p-value	Adj. significance level
OBV	**2.46E-06**	0.017 (0.05/3)
MACD	**1.94E-05**	0.025 (0.05/2)
MTM	**1.94E-05**	0.05 (0.05/1)

(b) Expected Rate of Return

Algorithm	p-value	Adj. significance level
OBV	**8.31E-13**	0.017 (0.05/3)
MTM	**4.52E-11**	0.025 (0.05/2)
MACD	**2.63E-07**	0.05 (0.05/1)

(c) Risk

Algorithm	p-value	Adj. significance level
OBV	**1.01E-13**	0.017 (0.05/3)
MACD	**1.29E-04**	0.025 (0.05/2)
MTM	**4.14E-04**	0.05 (0.05/1)

where the GP-DC-PT algorithm becomes 8.87% (0.12) and 10.69% (0.31). Last but not least, as indicated in Table 5.23, GP-DC-PT outperforms the buy-and-hold strategy in terms of the Sharpe ratio when considering both 5-year and 10-year datasets. In this case, the GP-DC-PT algorithm performs better than the buy-and-hold strategy.

Even so, we need to examine the statistical results. As before, the KS test is used. Since there is only one comparison between the GP-DC-PT algorithm and the buy-and-hold strategy, we did not apply the Holm-Bonferroni correction. The p-values between the GP-DC-PT algorithm and buy-and-hold strategy are 0.0012 and 4.14E-04, respectively, below 0.05 (the 5% significance level), indicating that the differences are statistically significant. Combined with the preceding discussion, we argue that the GP-DC-PT algorithm significantly outperforms the buy-and-hold strategy.

Lastly, it is worth noting that the GP-DC-PT algorithm could use the TR as the fitness function in lieu of the current Sharpe ratio. In this case, its average TR jumps to over 30%, which is higher than the buy-and-hold strategy's 27.21%.[3]

[3]Of course, since risk would no longer be part of the fitness function, the risk of GP-DC-PT would increase from 0.06 to 0.10. Nevertheless, this shows the flexibility of GP algorithms, which can easily adapt their fitness function to the different needs of traders.

Table 5.20: Kolmogorov-Smirnov tests between GP-DC-PT (control) and three technical indicators for TR, $\mathbb{E}[RoR]$, and risk on a 10-year period. p-values (second column) below the adjusted significance level (third column) appear in boldface to indicate statistical significance at the 5% level. The calculation of the adjusted significance level is shown in brackets in the third column.

(a) TR

Algorithm	p-value	Adj. significance level
OBV	**3.71E-05**	0.017 (0.05/3)
MACD	**0.0055**	0.025 (0.05/2)
MTM	**0.0055**	0.05 (0.05/1)

(b) Expected Rate of Return

Algorithm	p-value	Adj. significance level
MTM	**1.03E-08**	0.017 (0.05/3)
OBV	**5.65E-07**	0.025 (0.05/2)
MACD	**1.29E-04**	0.05 (0.05/1)

(c) Risk

Algorithm	p-value	Adj. significance level
OBV	**2.98E-10**	0.017 (0.05/3)
MACD	**7.22E-04**	0.025 (0.05/2)
MTM	**6.37E-12**	0.05 (0.05/1)

Table 5.21: Summary statistics between GP-DC-PT algorithm with buy-and-hold on the 5-year period. We use boldface for the best values for each measure.

	Total return	
Measurement	GP-DC-PT	buy-and-hold
Average	10.24%	**13.31**%
Median	**9.03**%	2.05%
Standard deviation	**0.18**	0.84
Maximum	130.98%	**755.68**%
Minimum	**-40.45**%	-82.85%

5.5 Conclusion

Our study explored the advantages of using genetic programming together with indicators based on the framework of DCs to create novel trading strategies. Our main achievement was conducting a thorough investigation of the performance of these trading strategies, conducting experiments on 220 datasets from 10 different markets, covering periods of 5 and 10 years.

Our investigation showed that both our proposed GP-DC and GP-DC-PT algorithms outperformed the baseline GP-PT in terms of total return (TR) and risk, with GP-DC-PT demonstrating the lowest overall risk across all tested datasets. The 5-year period saw GP-DC leading in TR and expected rate of return ($\mathbb{E}[RoR]$) while maintaining a lower risk when compared to

Table 5.22: Summary statistics between GP-DC-PT algorithm with buy-and-hold on the 10-year period. We use boldface for the best values for each measure.

	Total return	
Measurement	GP-DC-PT	buy-and-hold
Average	14.30%	**41.11**%
Median	10.85%	**11.44**%
Standard deviation	**0.38**	1.81
Maximum	186.62%	**1753.05**%
Minimum	-229.56%	**-89.62**%

Table 5.23: Kolmogorov-Smirnov tests between GP-DC-PT (control) and buy-and-hold strategy for TR, $\mathbb{E}[RoR]$, and risk on a 10-year period. p-values below 0.05 appear in boldface to indicate statistical significance at the 5% level.

Algorithm	p-value
Buy-and-hold (5 years)	**0.0012**
Buy-and-hold (10 years)	**4.14E-04**

GP-DC-PT. However, GP-DC-PT proved its consistency by achieving the highest Sharpe ratio across both 5 and 10-year periods, even when GP-PT yielded the highest average TR in the 10-year analysis. Notably, even the best-performing models showed that GP-DC-PT could surpass both GP-DC and GP-PT in TR and $\mathbb{E}[RoR]$, albeit with slightly higher risk as compared to GP-DC for these top models. Furthermore, GP-DC-PT successfully outperformed the nine ML algorithms in terms of TR and $\mathbb{E}[RoR]$, achieving the expected goal.

When GP-DC-PT was compared to technical analysis trading strategies, it significantly outperformed three commonly used technical indicators (MACD, OBV, MTM) in terms of return, risk, and the Sharpe ratio. GP-DC-PT also showed a better performance of TR across both 5- and 10-year periods when compared to buy-and-hold.

In conclusion, our findings highlight the effectiveness of combining a GP with DC-based indicators. This approach yielded profitable results at low risk, outperforming traditional physical time strategies such as buy-and-hold. Both GP-DC and GP-DC-PT achieved a median TR of around 10% across a large dataset, with remarkably low risk levels. Additionally, GP-DC-PT consistently ranked first in terms of the Sharpe ratio, followed by GP-DC, across different time frames. Their ability to outperform GP-PT across various tests further strengthens

their potential.

While the above results are very positive, they were all under the aggregate fitness function of the Sharpe ratio. A disadvantage of such fitness functions is that they do not look into each component (i.e. expected rate of return, risk) separately. This can lead to certain 'undesirable' behaviours, e.g. evolving passive trees with only two trades throughout the whole trading period so that the standard deviation (the denominator in the Sharpe ratio formula) becomes very small and, as a result, the Sharpe ratio becomes very large. To avoid such behaviours, in the following chapter, we propose using a multi-objective optimisation fitness function.

Chapter 6

Genetic programming application on multi-objective optimisation

6.1 Motivation

This chapter extends the work presented in Chapter 5 by using a multi-objective optimisation fitness function for GP-DC-PT, which has, so far, been the best-performing algorithm. However, a behaviour often observed during our experiments was that the GP would show a preference for individuals favouring passive trading, i.e. very few trades during the trading period. This would result in a very small standard deviation (risk) value. Indeed, as we observed in Chapter 5, GP-DC-PT often showed the best risk performance as compared to other algorithms. This low risk could frequently be attributed to a low number of trades. On the other hand, because the Sharpe ratio is an aggregate fitness function, the GP would also completely disregard the expected RoR performance (the numerator of the Sharpe ratio), as it was already achieving extremely high fitness values due to the low risk. While the above behaviour fulfils the purpose of maximizing the Sharpe ratio, it would not be appropriate in real-life trading as, essentially, it would yield a relatively low expected RoR.

To overcome this behaviour, we propose using the NSGA-II algorithm to optimise the expected RoR and risk simultaneously, rather than as part of an aggregate fitness function. This will allow the GP algorithm to place emphasis on both metrics during search and produce

trading strategies that yield high returns while maintaining low-risk values. The NSGA-II is applied to the GP-DC-PT, which was the best performer in the previous chapter. This is the first proposed algorithm in this chapter, which we call MOO2 since it is a multi-objective optimisation algorithm which optimises two metrics. In addition, we extend MOO2 by applying the α-dominance strategy presented in Chapter 2. As previously explained, NSGA-II applies strict Pareto dominance criteria, meaning a solution is considered better than another only if it improves on at least one objective without worsening any others. This strict criterion can however lead to undesired behaviour, as a solution with a strong performance in one objective but poor performance in others can persist through the NSGA-II generations[1]. On the other hand, the α-dominance strategy relaxes the strict Pareto dominance criteria by adding a coefficient that considers other objectives when comparing one objective between two solutions. This allows for a more balanced evaluation of solutions across the two objectives of MOO2. We call this extended algorithm αMOO2. Lastly, we propose another multi-objective optimisation algorithm we call MOO3, which optimizes total return, the expected RoR, and risk. The reason for adding the additional objective of total return is so that long-term profitability is also taken into account. We do not need to apply α-dominance to MOO3, because the problem that the strict dominance caused was not observed in the experiments of the MOO3. This could be explained by the additional fitness function total return (TR) in MOO3. The local optimal, which was reached by MOO2, is overcome by MOO3 when considering TR. In other words, the conflict between $\mathbb{E}[RoR]$ and risk is relieved when optimising TR, $\mathbb{E}[RoR]$, and risk simultaneously, which is an interesting finding. However, since it is not the main focus of this thesis, we will defer this issue to future work.

The rest of this chapter is organised as follows. Section 6.2 presents the methodology used in this chapter, then Section 6.3 discusses the benchmarks in our experiments, while Section 6.4 presents and discusses the results. Finally, Section 6.5 concludes this chapter.

[1]NSGA-II has parents compete with offspring, retaining the top half of solutions with better performance. This ensures that a solution with strong performance in one objective will always survive, as no other solutions can dominate it.

6.2 Methodology

As mentioned above, in this chapter we will be using the GP-DC-PT algorithm with a multi-objective optimisation fitness function. The end goal of the algorithm is to evolve trading strategies that are optimal in the multi-objective sense. Specifically, we aim to discover strategies that achieve a desirable balance between the two (expected RoR and risk) or three (total return, expected RoR, and risk) key objectives. To this end, we employ NSGA-II [38] as the main GP evolution strategy. Once the set of suitable, Pareto-optimal solutions has been obtained via this process, we make a final, singular choice from within that set, by using a predefined criterion designed to reflect trader preference (i.e. the desired balance among the various objectives). The sections that follow explore the different parts of the process described above in more detail.

6.2.1 Selection and genetic operators for MOO2, αMOO2, and MOO3

In the GP process, the crossover and mutation are applied to simulate the natural evolutionary processes. To decide which individuals will be used in crossover or mutation, a selection method is employed. In this thesis, we use the tournament as the GP selection method, as discussed in Section 4.2.1.4. The tournament selection first chooses randomly a certain number of individuals and then the fitness function of these individuals is compared. The individual with the best fitness function will be considered the winner and progress to the crossover or mutation. However, in the NSGA-II, multiple objectives need to be optimised, making the direct comparison of fitness function impossible. Thus to select the winner of a tournament, NSGA-II considers the Pareto front rank and the crowding distance instead of the fitness functions. After obtaining a certain number of individuals for the tournament, NSGA-II first compares the Pareto front rank—the individual with a lower rank survives. For individuals with equal Pareto front rank, the one with a higher crowding distance is selected.

With regards to genetic operators, we still use subtree crossover and point mutation.

6.2.2 α-dominance strategy αMOO2

As previously mentioned, to overcome the problem that solutions with extremely poor performance in one objective remain in the later generations of the NSGA-II, caused by Pareto's strict dominance criterion, we use the α-dominance strategy, as introduced in Equation 2.21. The proposed MOO GP algorithm has two objectives, namely the expected rate of return ($\mathbb{E}[RoR]$) and risk. Hence, the objectives i and j from the original Equation 2.21 now become $\mathbb{E}[RoR]$ and risk, respectively. In addition, because the two objectives are conflicting (one maximisation, one minimisation), the sign in front of the α parameter from Equation 2.21 needs to change from positive (+) to negative (-). Hence, the α-dominance strategy for expected RoR is:

$$G_{RoR}(A,B) = f_{RoR}(A) - f_{RoR}(B) - \alpha_{RoR,Risk}(f_{Risk}(A) - f_{Risk}(B)) \tag{6.1}$$

Similarly, the α-dominance strategy for risk is:

$$G_{Risk}(A,B) = f_{Risk}(A) - f_{Risk}(B) - \alpha_{Risk,RoR}(f_{RoR}(A) - f_{RoR}(B)) \tag{6.2}$$

Further, as mentioned in Section 2.3.2, there are three adaptation schemes to determine the value of $\alpha_{i,j}$, namely f_{linear}, $f_{sigmoid}$, and f_{cosine} (see Equation 2.22). In our experiments, we apply each one of these three adaption schemes to Equations 6.1 and 6.2. Hence, we end up with six different α-dominance strategy configurations: (i) $\mathbb{E}[RoR]$ with sigmoid adaptation ($\alpha MOO2^{Sig}_{RoR}$); (ii) $\mathbb{E}[RoR]$ with cosine adaptation ($\alpha MOO2^{Cos}_{RoR}$); (iii) $\mathbb{E}[RoR]$ with linear adaptation ($\alpha MOO2^{Lin}_{RoR}$); (iv) Risk with sigmoid adaptation ($\alpha MOO2^{Sig}_{Risk}$); (v) Risk with cosine adaptation ($\alpha MOO2^{Cos}_{Risk}$); and, (vi) Risk with linear adaptation ($\alpha MOO2^{Lin}_{Risk}$). 'MOO2' refers to the fact that each algorithm is a two-objective optimisation algorithm. It is worth noting that we only include the α-dominance strategy in MOO2 because the results for MOO3 were not satisfactory.

6.2.3 Designating the final solution

In our experiments, we will present summary statistics for the single and MOO GP algorithms over 50 independent runs. The output of the NSGA-II at the end of this process is not a single solution, but rather an entire front of solutions that are optimal in the Pareto sense. While obtaining such a Pareto front is a desired property of any multi-objective algorithm in theory, in practice it is useful to designate a single, definitive choice from that front, as the desired, most representative solution resulting from this process. This serves at least two purposes. Firstly, it provides a single, final solution (i.e., trading strategy), whose performance or quality can be evaluated and compared directly against other trading algorithms or strategies serving as benchmarks. Secondly, it allows the trader to select the 'best' solution from the set, according to some criteria valued by the trader, namely the differential extent to which they value each of the multiple objectives. Note that prior to this point in the process, such a preference would not have played any part in the genetic process.

Since we propose MOO2 and MOO3, we have different ways to identify the 'best' solutions. When we optimise the two fitness functions to select the best model from SOO, we look into the 50 independent runs in training, find the best model (in terms of Sharpe ratio), and report its test set value. With regards to the MOO2 algorithms, although we cannot strictly talk about a 'best' solution on the Pareto front, we needed to define a way of selecting a single solution to compare it with the SOO. Given that the two objectives of MOO2 are the expected RoR and risk, which are also the components of SOO's fitness function of Sharpe ratio, we decided to look into the 50 independent GP runs' *Pareto fronts* from the final generation in the training set and select the model with the best Sharpe ratio. We then report the model's Sharpe ratio in the test set.

When considering the three fitness functions (TR, $\mathbb{E}[RoR]$, and Risk), we also need to compare the SOO and MOO3 algorithms in terms of a single 'best' solution. In order to do this, we define a new aggregate metric, which effectively acts as a generalisation of the Sharpe ratio that is able to take into account total return as well as the expected rate of return and risk. In order to be able to calculate the metric, we first need to normalise the three objectives into the range $[0, 1]$; we do so by considering the range of values resulting for each objective

in the final Pareto front. This 'modified Sharpe Ratio' (mSR) metric is then defined as per Equation 6.3:

$$mSR = \frac{(\widehat{\mathsf{TR}} + 1)^a \times (\widehat{\mathbb{E}[\mathsf{RoR}]} + 1)^b}{(\widehat{\mathsf{Risk}} + 1)^c} \tag{6.3}$$

where $\widehat{TR}$ is the normalised total return value of the particular individual, $\widehat{\mathbb{E}[RoR]}$ is its normalised expected rate of return, $\widehat{Risk}$ is its normalised risk value, while a, b, and c are weights that determine the importance of each term in the metric, whereupon we impose the additional constraint here that $a + b + c = 1$. It is necessary to add 1 to each normalised value here so that each term becomes monotonically non-decreasing as its corresponding weight increases. Under this scheme, each term can achieve a minimum value of 1, and a maximum value of 2. By adjusting the values of a, b, and c, we can control the emphasis placed on each metric within the GP process. For example, a term with zero weight will achieve the minimum value of 1, thus having no influence in the end score; a term with a maximum weight of 1, results in that term dominating the calculation and all other terms being ignored due to the constraint. In the special case where b and c are equal to 0.5 (and thus $a = 0$), the metric disregards total return, effectively reducing to the square root of the Sharpe ratio (assuming normalised values).

It is important to note here that, due to the normalisation requirement, the above metric primarily serves as a way of scoring solutions within a given Pareto front (or population of solutions more generally); it is not a general aggregate metric that can be used to evaluate individual solutions directly outside of a population context, such as in the case of the original Sharpe Ratio. If we *did* want to use this metric to compare against benchmark solutions, these would first need to have their objectives normalised within the same range, as dictated by the Pareto front of interest.

6.3 Experimental set-up

In this section, we present what benchmarks we used to compare our proposed approach in Section 6.3.1. The datasets and parameter tuning follow the same process presented in Section 4.3.1 and Section 5.3.2.

6.3.1 Benchmarks

6.3.1.1 MOO2

We benchmark the MOO2 algorithms against a single-objective optimisation (SOO) GP-based approach that employs the Sharpe ratio as its fitness function.

Furthermore, we benchmark the MOO2 algorithms against trading strategies derived from three popular technical analysis indicators: MACD, OBV, and MTM, as introduced in Section 5.3.1.

Lastly, we also compare the proposed MOO2 algorithms against the passive trading strategy of buy-and-hold.

6.3.1.2 MOO3

To benchmark MOO3, we also compare it to a SOO GP-based approach. As mentioned in Section 6.2.3, the fitness function we use is the aggregate function presented in Equation 6.3. We designed the seven SOOs following seven setups listed in Section 6.4.2.1.

Further, we benchmark MOO3 against the three technical indicators MACD, OBV, and MTM, together with the buy-and-hold strategy.

6.4 Results and analysis

We run all GP algorithms for 50 independent runs and report the results below.

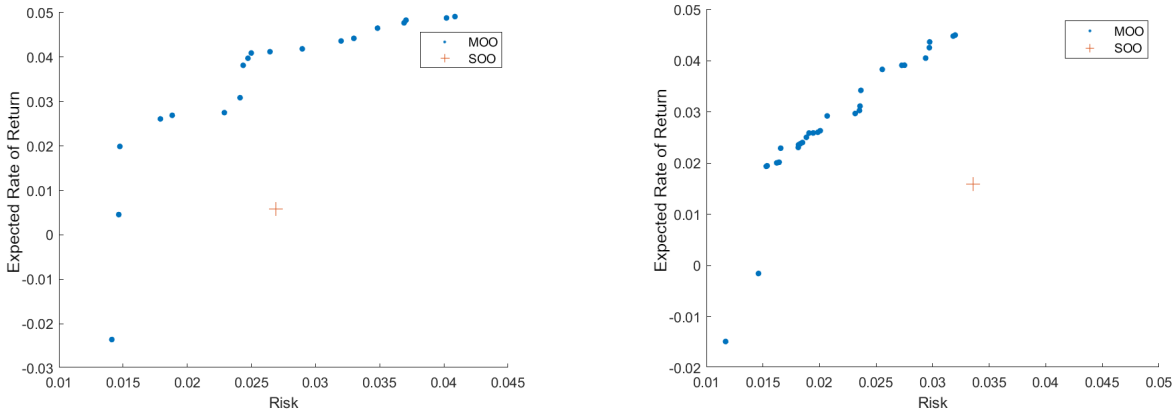
In this section, we report and analyse the performance of MOO2 and MOO3 against the benchmark approaches. We first present the results for MOO2 (Section 6.4.1) and then

present the results for MOO3 (Section 6.4.2). Lastly, we present a comparison of MOO2 and MOO3 algorithms (Section 6.4.3).

6.4.1 MOO2

6.4.1.1 Pareto front

Figure 6.1 shows the Pareto front for the $\alpha MOO2^{cos}_{RoR}$ algorithm, along with the best model (highest Sharpe ratio) from the SOO GP, which was selected across the 50 independent GP runs. The $\alpha MOO2^{cos}_{RoR}$ algorithm was selected as a representative example, and the results are similar to the other multi-objective GPs. As we can observe from the two plots, there are several solutions on the Pareto front. For Accenture (Figure 6.1a), the majority of solutions have a higher expected rate of return than the one by SOO. Several solutions dominate the SOO model, as they have both a higher return and lower risk. For Experian (Figure 6.1b) the results are even better for $\alpha MOO2^{cos}_{RoR}$, as the majority of its solutions dominate the SOO solution.



(a) Accenture plc (S&P500) (b) Experian plc (FTSE100)

Figure 6.1: Comparison of the Pareto front solutions with the best single objective optimisation GP.

6.4.1.2 Comparison of MOO2 to corresponding SOO approaches

Although MOO algorithms are able to find solutions in the Pareto front that dominate solutions from the SOO GP, in the real world a trader would be interested in identifying a *single*

trading strategy to use in the market. As Section 6.2.3 introduced, this section compares the best model derived from SOO and the 'best' model obtained from the MOO2 algorithms.

Table 6.1 presents the performance of the SOO, MOO2, and $\alpha MOO2$ algorithms over a 5-year period. We can observe that $\alpha MOO2^{Sig}_{RoR}$ achieves the best average (1.82%) and median (1.66%) value for $\mathbb{E}[RoR]$, while SOO observes the worst average (1.13%) and median (1.17%) values. It is worth noting that MOO2 also yields the same median $\mathbb{E}[RoR]$ values as SOO but achieves a better average $\mathbb{E}[RoR]$ value (1.66%) ranked second across all algorithms. Regarding risk (middle part of Table 6.1), SOO is the top performer, followed by the $\alpha MOO2$ algorithms, which exhibit similar risk levels; MOO2 has slightly worse risk figures across all statistics. Lastly, the bottom part of the table presents the Sharpe ratio of the portfolio of 110 stocks (assuming equal weight) per algorithm. Note that this is a single value for the portfolio and hence no summary statistics are presented for the Sharpe ratio. As we observe, $\alpha MOO2^{Sig}_{Risk}$ has the highest value (0.83), followed by $\alpha MOO2^{Sig}_{RoR}$.

Table 6.2 presents the performance of the SOO, MOO2, and $\alpha MOO2$ algorithms over a 10-year period. We can observe that $\alpha MOO2^{RoR}_{Cos}$ achieves the best average (1.84%) and median (1.68%) value for the expected RoR, while SOO observes the worst average (0.91%) and median (0.77%) values. Regarding risk (middle part of Table 6.2), the performance is the same for the 5 years. Lastly, in terms of the Sharpe ratio metric, $\alpha MOO2^{RoR}_{Sig}$ has the highest value (1.11). This figure is 0.4 higher than the benchmark SOO algorithm.

We also conduct pairwise comparisons between SOO and the MOO2 algorithms using the Kolmogorov-Smirnov (KS) non-parametric statistical test to assess the significance of the observed results. Our focus was on understanding the enhancements brought about by each MOO2 algorithm over the SOO. The null hypothesis is that the two distributions originate from the same continuous distribution. Given the seven pairwise comparisons, we again apply the Holm-Bonferroni correction to account for these multiple comparisons.

Table 6.3 and Table 6.4 present the KS test results for the expected RoR and risk. As observed, all MOO algorithms statistically and significantly outperform the SOO's expected RoR over a 10-year period. In 5-year data, there is no significant difference observed. In terms of risk, MOO2's poor performance is statistically outperformed by SOO over both 5

and 10-year periods. No other statistical differences between the αMOO2 algorithms and SOO are observed, which confirms our earlier observation that the $\alpha MOO2$ algorithms have a similar risk performance to SOO. We can thus conclude that the $\alpha MOO2$ algorithms have statistically and significantly improved the expected rate of return over the SOO algorithm while maintaining similar risk levels for the 10-year period.

Table 6.1: Comparison between SOO and MOO2 algorithms on 5-year periods. Best values per row appear in boldface.

Measurement	Expected Rate of Return							
Algorithm	SOO	MOO2	$\alpha MOO2^{Sig}_{Risk}$	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{Risk}$	$\alpha MOO2^{Cos}_{RoR}$	$\alpha MOO2^{Lin}_{Risk}$	$\alpha MOO2^{Lin}_{RoR}$
Average	1.13%	1.66%	1.62%	**1.82**%	1.28%	1.58%	1.35%	1.65%
Median	1.17%	1.19%	1.42%	**1.66**%	1.27%	1.37%	1.21%	1.50%
Standard deviation	0.02	0.03	0.02	0.02	0.02	0.02	0.02	0.02
Max	7.74%	**19.95**%	9.88%	12.06%	8.55%	11.24%	7.38%	11.73%
Min	**-3.82**%	-6.60%	-7.24%	-4.32%	-8.91%	-8.10%	-4.86%	-6.53%
Measurement	Risk							
Algorithm	SOO	MOO2	$\alpha MOO2^{Sig}_{Risk}$	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{Risk}$	$\alpha MOO2^{Cos}_{RoR}$	$\alpha MOO2^{Lin}_{Risk}$	$\alpha MOO2^{Lin}_{RoR}$
Average	**0.06**	0.10	0.07	0.08	0.07	0.07	0.07	0.08
Median	**0.05**	0.08	**0.05**	0.06	0.06	0.06	0.06	0.07
Standard deviation	**0.04**	0.07	0.05	0.05	0.05	0.05	**0.04**	0.05
Max	**0.22**	0.55	0.32	0.34	0.38	0.33	0.25	0.35
Min	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Measurement	Sharpe ratio							
Algorithm	SOO	MOO2	$\alpha MOO2^{Sig}_{Risk}$	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{Risk}$	$\alpha MOO2^{Cos}_{RoR}$	$\alpha MOO2^{Lin}_{Risk}$	$\alpha MOO2^{Lin}_{RoR}$
	0.72	0.82	**0.83**	0.82	0.63	0.67	0.73	0.71

Table 6.2: Comparison between SOO and MOO2 algorithms on 10-year periods. Best values per row appear in boldface.

Measurement	Expected Rate of Return							
Algorithm	SOO	MOO2	$\alpha MOO2^{Sig}_{Risk}$	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{Risk}$	$\alpha MOO2^{Cos}_{RoR}$	$\alpha MOO2^{Lin}_{Risk}$	$\alpha MOO2^{Lin}_{RoR}$
Average	0.91%	1.64%	1.65%	1.67%	1.81%	**1.84**%	1.49%	1.65%
Median	0.77%	1.55%	1.66%	1.66%	1.55%	**1.68**%	1.46%	1.50%
Standard deviation	**0.01**	0.04	0.02	0.01	0.02	0.02	0.02	0.02
Max	6.00%	**29.91**%	8.22%	7.98%	11.35%	7.21%	8.28%	9.24%
Min	-3.33%	-6.04%	-5.22%	-**1.91**%	-2.34%	-2.18%	-2.54%	-4.45%
Measurement	Risk							
Algorithm	SOO	MOO2	$\alpha MOO2^{Sig}_{Risk}$	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{Risk}$	$\alpha MOO2^{Cos}_{RoR}$	$\alpha MOO2^{Lin}_{Risk}$	$\alpha MOO2^{Lin}_{RoR}$
Average	**0.06**	0.09	0.07	0.07	0.07	0.07	0.07	0.07
Median	**0.05**	0.07	0.06	0.06	0.06	0.06	0.06	0.06
Standard deviation	**0.04**	0.06	0.05	0.05	**0.04**	0.05	**0.04**	0.06
Max	**0.24**	0.45	0.32	0.33	0.28	0.29	**0.24**	0.46
Min	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Measurement	Sharpe ratio							
Algorithm	SOO	MOO2	$\alpha MOO2^{Sig}_{Risk}$	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{Risk}$	$\alpha MOO2^{Cos}_{RoR}$	$\alpha MOO2^{Lin}_{Risk}$	$\alpha MOO2^{Lin}_{RoR}$
	0.70	0.92	0.93	**1.11**	0.86	1.08	0.91	0.81

Table 6.3: Kolmogorov-Smirnov tests between SOO (control) and MOO2 algorithms (first column) for $\mathbb{E}[RoR]$ and risk over a 5-year period. p-values (second column) below the adjusted significance level (third column) appear in boldface to indicate statistical significance at the 5% level. The calculation of the adjusted significance level is shown in brackets in the third column.

(a) Expected Rate of Return

Algorithm	p-value	Adj. significance level
MOO2	0.0461	0.0071 (0.05/7)
$\alpha MOO2^{Sig}_{RoR}$	0.0944	0.0083 (0.05/6)
$\alpha MOO2^{Lin}_{RoR}$	0.3141	0.010 (0.05/5)
$\alpha MOO2^{Cos}_{RoR}$	0.5060	0.013 (0.05/4)
$\alpha MOO2^{Sig}_{Risk}$	0.5060	0.017 (0.05/3)
$\alpha MOO2^{Lin}_{Risk}$	0.7336	0.025 (0.05/2)
$\alpha MOO2^{Cos}_{Risk}$	0.9237	0.05 (0.05/1)

(b) Risk

Algorithm	p-value	Adj. significance level
MOO2	**5.65E-07**	0.0071 (0.05/7)
$\alpha MOO2^{Lin}_{RoR}$	0.1790	0.0083 (0.05/6)
$\alpha MOO2^{Cos}_{RoR}$	0.2395	0.010 (0.05/5)
$\alpha MOO2^{Sig}_{RoR}$	0.3141	0.013 (0.05/4)
$\alpha MOO2^{Lin}_{Risk}$	0.4033	0.017 (0.05/3)
$\alpha MOO2^{Cos}_{Risk}$	0.5060	0.025 (0.05/2)
$\alpha MOO2^{Sig}_{RoR}$	0.9998	0.05 (0.05/1)

Table 6.4: Kolmogorov-Smirnov tests between SOO (control) and MOO2 algorithms (first column) for $\mathbb{E}[RoR]$ and risk over a 10-year period. p-values (second column) below the adjusted significance level (third column) appear in boldface to indicate statistical significance at the 5% level. The calculation of the adjusted significance level is shown in brackets in the third column.

(a) Expected Rate of Return

Algorithm	p-value	Adj. significance level
$\alpha MOO2^{Sig}_{RoR}$	**1.29E-04**	0.0071 (0.05/7)
$\alpha MOO2^{Cos}_{RoR}$	**1.29E-04**	0.0083 (0.05/6)
$\alpha MOO2^{Sig}_{RoR}$	**4.14E-04**	0.010 (0.05/5)
$\alpha MOO2^{Lin}_{RoR}$	**7.22E-04**	0.013 (0.05/4)
MOO2	**0.0034**	0.017 (0.05/3)
$\alpha MOO2^{Cos}_{Risk}$	**0.0055**	0.025 (0.05/2)
$\alpha MOO2^{Lin}_{Risk}$	**0.0137**	0.05 (0.05/1)

(b) Risk

Algorithm	p-value	Adj. significance level
MOO2	**2.33E-04**	0.0071 (0.05/7)
$\alpha MOO2^{Sig}_{RoR}$	0.0461	0.0083 (0.05/6)
$\alpha MOO2^{Lin}_{RoR}$	0.0461	0.010 (0.05/5)
$\alpha MOO2^{Sig}_{RoR}$	0.0944	0.013 (0.05/4)
$\alpha MOO2^{Cos}_{Risk}$	0.0944	0.017 (0.05/3)
$\alpha MOO2^{Lin}_{Risk}$	0.0944	0.025 (0.05/2)
$\alpha MOO2^{Cos}_{RoR}$	0.1312	0.05 (0.05/1)

6.4.1.3 Comparison of MOO2 algorithms to technical analysis trading strategies

In this section, we are interested in selecting the best performing $\alpha MOO2$ algorithms to bring forward for comparison with technical analysis indicators. Looking back at Table 6.2, we can observe that $\alpha MOO2^{RoR}_{Cos}$ and $\alpha MOO2^{RoR}_{Sig}$ had the highest average and median expected rate of return in 5 and 10-year periods, separately. In terms of risk, all algorithms showed very similar performance. Given that the above two algorithms had the best performance in terms of the expected RoR, we will use both in the comparisons with trading strategies derived from technical indicators.

Table 6.5 and Table 6.6 compare the performance of the two MOO2 algorithms and the

technical indicators in 5 and 10-year periods. Both $\alpha MOO2^{Sig}_{RoR}$ and $\alpha MOO2^{Cos}_{RoR}$ have high average and median expected RoR values for both periods. MACD appears to be competitive when looking at its average values (1.31% and 1.71%), but this is due to outliers; its median values are only 0% and 0.14% over the 5- and 10-year periods, respectively. The two $\alpha MOO2$ algorithms also have the lowest risk values for both average (0.07) and median (0.06), which are almost 50% lower than the risk values of MACD, OBV, and MTM over the 10-year period. Over the 5-year period, however, the OBV indicator and $\alpha MOO2^{RoR}_{Cos}$ are the top performers, both having the same average and median risk values. This could be explained by the sacrifice of OBV's negative average RoR values. Lastly, $\alpha MOO2^{Sig}_{RoR}$ has the highest Sharpe ratio value (0.82 and 1.11), closely followed by $\alpha MOO2^{Cos}_{RoR}$, while the highest value by a technical indicator is MACD (0.19 and 0.26) across 5 and 10-year periods, separately.

Table 6.5: Summary statistics of the best two $\alpha MOO2$ algorithms and technical indicators over a 5-year period. Best value per row is denoted in boldface.

	Expected Rate of Return				
	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{RoR}$	MACD	OBV	MTM
Average	**1.82**%	1.58%	1.31%	-0.03%	0.32%
Median	**1.66**%	1.37%	0.00%	0.00%	0.02%
StDev	**0.02**	0.02	0.07	0.05	**0.02**
Max	12.06%	11.24%	**31.84**%	21.52%	10.98%
Min	-4.32%	-8.10%	-24.31%	-24.60%	**-2.59**%
	Risk				
	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{RoR}$	MACD	OBV	MTM
Average	0.08	**0.07**	0.09	**0.07**	0.12
Median	0.06	0.06	**0.05**	**0.05**	0.08
StDev	**0.05**	**0.05**	0.15	0.07	0.15
Max	0.34	**0.33**	1.05	0.35	0.95
Min	0.01	0.01	0.01	0.01	0.02
	Sharpe ratio				
	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{RoR}$	MACD	OBV	MTM
	0.82	0.67	0.19	-0.01	0.17

To compare the performance among the different algorithms (rather than performing pair-

Table 6.6: Summary statistics of the best two $\alpha MOO2$ algorithms and technical indicators. Best value per row is denoted in boldface.

	Expected Rate of Return				
	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{RoR}$	MACD	OBV	MTM
Average	1.67%	**1.84**%	1.71%	0.77%	0.32%
Median	1.66%	**1.68**%	0.14%	0.18%	0.11%
StDev	0.01	0.02	0.06	0.05	**0.01**
Max	7.98%	7.21%	36.01%	**35.65**%	6.95%
Min	-1.91%	-2.18%	-9.04%	-11.11%	**-1.48**%
	Risk				
	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{RoR}$	MACD	OBV	MTM
Average	**0.07**	**0.07**	0.12	0.12	0.15
Median	**0.06**	**0.06**	0.07	0.09	0.09
StDev	**0.05**	**0.05**	0.22	0.12	0.20
Max	0.33	**0.29**	1.58	0.76	1.82
Min	0.01	0.01	0.01	0.01	0.03
	Sharpe ratio				
	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{RoR}$	MACD	OBV	MTM
	1.11	1.08	0.26	0.15	0.29

wise comparisons as earlier)[2] we ran the Friedman non-parametric test, where we calculate the average rank of each algorithm in terms of return and risk—the lower the rank, the better the algorithm's performance. We also perform the Bonferroni post-hoc test and present both in Table 6.7. For each algorithm, the table shows the average rank (first column), and the adjusted p-value of the statistical test when that algorithm's average rank is compared to the average rank of the algorithm with the best rank (control algorithm) according to Bonferroni's post-hoc test (second column) [155, 156]. As we can observe, in terms of both return and risk $\alpha MOO2^{Cos}_{RoR}$ ranks first and statistically outperforms all technical indicator strategies.

6.4.1.4 Buy-and-hold

We now compare the two $\alpha MOO2$ algorithms with the buy-and-hold strategy. Table 6.10 and Table 6.11 present the performance metrics of the MOO2 algorithms alongside the buy-and-

[2]In Section 6.4.1.2, we were interested in reporting how each MOO2 algorithm compared to the SOO GP. We thus used pairwise Kolmogorov-Smirnov comparisons. On the other hand, when comparing the three technical indicator results, we were more interested in identifying the best algorithm. For this reason, we resorted to the Friedman test.

Table 6.7: Friedman with Bonferroni's post-hoc test between $\alpha MOO2$ and technical indicators. Statistically significant differences at 5% level shown in boldface.

Table 6.8: Expected Rate of Return

Algorithm	Average rank	p_{Bonf}
$\alpha MOO2^{Cos}_{RoR}$ (c)	2.23	-
$\alpha MOO2^{Sig}_{RoR}$	2.41	0.19
MACD	3.31	**7.45E-07**
OBV	3.43	**6.88E-08**
MTM	3.60	**1.76E-09**

Table 6.9: Risk

Algorithm	Average rank	p_{Bonf}
$\alpha MOO2^{Cos}_{RoR}$ (c)	2.24	-
$\alpha MOO2^{Sig}_{RoR}$	2.30	0.07
MACD	2.86	**5.70E-04**
OBV	3.55	**6.60E-10**
MTM	4.03	**7.09E-16**

hold strategy over 5 and 10-year periods. Over the 10-year dataset, the buy-and-hold has the highest average value, but this is only due to outliers. When looking at the median, the two MOO2 algorithms have around a 6% higher total return. Over the 5-year dataset, the $\alpha MOO2^{Sig}_{RoR}$ has the highest average and median TR value. We again use the non-parametric Friedman test to support the analysis. $\alpha MOO2^{Sig}_{RoR}$ and $\alpha MOO2^{Cos}_{RoR}$ rank first in the 5 and 10-year periods, respectively. Statistically, both of them significantly outperformed buy-and-hold, as can be seen from Table 6.12 and Table 6.13.

Table 6.10: Summary statistics for the best two MOO2 algorithms and buy-and-hold in terms of total return on the 5-year period. Best value per row appears in boldface.

Algorithm	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{RoR}$	Buy-and-hold
Average	**15.81**%	14.78%	13.31%
Median	**13.72%**	11.76%	2.05%
Standard deviation	**0.23**	0.28	0.84
Max	155.12%	181.55%	**755.68**%
Min	**-38.93%**	-64.20%	-82.85%

Table 6.11: Summary statistics for the best two MOO2 algorithms and buy-and-hold in terms of total return over the 10-year period. The best value per row appears in boldface.

Algorithm	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{RoR}$	Buy-and-hold
Average	30.04%	32.50%	**41.11%**
Median	17.33%	**19.02**%	11.44%
Standard deviation	0.52	**0.46**	1.81
Max	352.30%	294.70%	**1753.05**%
Min	-38.16%	**-31.20**%	-89.62%

Table 6.12: Friedman test with Bonferroni's post-hoc test between $\alpha MOO2$ and buy-and-hold over the 5-year period. Statistically significant differences at the 5% level are shown in boldface.

Algorithm	Average rank	p_{Bonf}
$\alpha MOO2^{Sig}_{RoR}$ (c)	1.72	-
$\alpha MOO2^{Cos}_{RoR}$	1.91	0.053
Buy-and-hold	2.37	**2E-06**

Table 6.13: Friedman test with Bonferroni's post-hoc test between $\alpha MOO2$ and buy-and-hold over the 10-year period. Statistically significant differences at the 5% level are shown in boldface.

Algorithm	Average rank	p_{Bonf}
$\alpha MOO2^{Cos}_{RoR}$ (c)	1.83	-
$\alpha MOO2^{Sig}_{RoR}$	1.92	0.16
Buy-and-hold	2.23	**0.003**

6.4.2 MOO3

6.4.2.1 Trader preference scenarios

As mentioned in section 6.2.3, while a multi-objective optimisation framework is necessary and beneficial to obtain solutions that optimise conflicting objectives, in the end, a trader will decide upon a single trading strategy. Therefore it is important to be able to designate a single solution from the Pareto front, in a manner that allows the trader to specify their preference over the various objectives. We do this here via the modified Sharpe Ratio, as introduced in Equation 6.3, by modifying the different weight values (a, b, c), where a corresponds to the weight given to total return, b to the expected rate of return, and c to risk. In our experiments, we have focused on seven different scenarios/set-ups, namely:

(i) $[a = 0.5, b = 0.5, c = 0]$. The fitness function focuses equally on total return and expected RoR.

(ii) $[a = 0, b = 0.5, c = 0.5]$. The fitness function focuses equally on an expected RoR and risk.

(iii) $[a = 0.5, b = 0, c = 0.5]$. The fitness function focuses equally on total return and risk.

(iv) $[a = 0.33, b = 0.33, c = 0.33]$. The fitness function focuses equally on all three metrics.

(v) $[a = 1, b = 0, c = 0]$. The fitness function focuses only on total return.

(vi) $[a = 0, b = 1, c = 0]$. The fitness function focuses only on the expected RoR.

(vii) $[a = 0, b = 0, c = 1]$. The fitness function focuses only on risk.

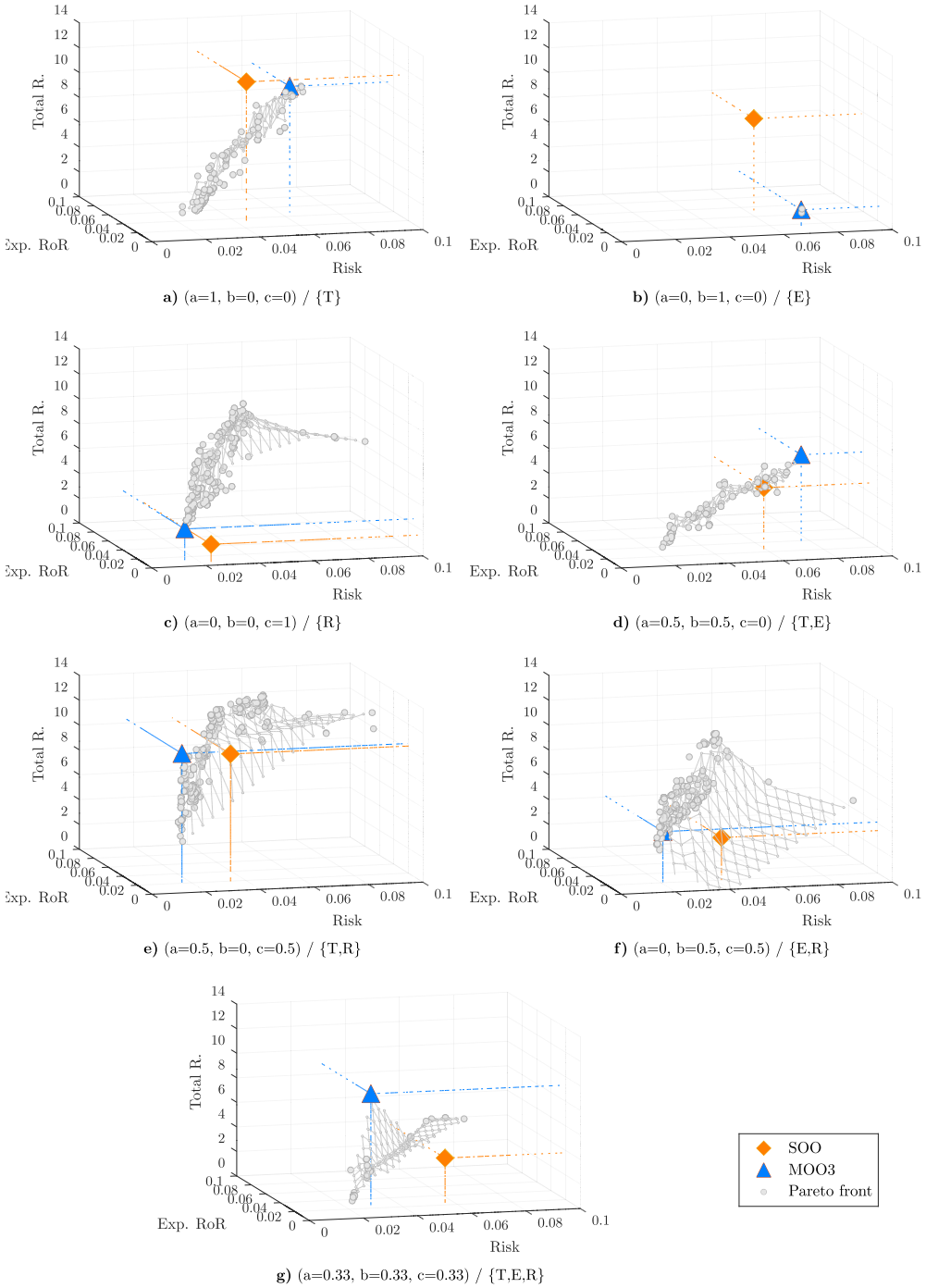
These set-ups allow us to consider different extreme cases, where only one or two metrics are being considered and also evaluate cases where all three metrics are equally being considered simultaneously.

6.4.2.2 Pareto front

We begin by presenting the results of the MOO3 algorithm and examining the placement of the MOO3-designated solutions *(and the Pareto front solutions more generally)* relative to the SOO solutions for each of the seven scenarios considered. For illustration, we use a single GP run for each scenario, focusing on the Apple stock. Figure 6.2 displays the generated solutions and resulting interpolated surface making up the Pareto front (grey mesh/markers); the designated MOO3-solution from that front (blue triangle marker); and, the corresponding SOO solution (orange diamond marker) for each of the seven scenarios. The MOO3 and SOO solutions are accompanied by 2D projections to aid interpretation and comparison. Each scenario is identified via its $[a, b, c]$ weight-triplet and, to enhance clarity, each triplet is also expressed as the corresponding subset of the set {T,E,R}, where the choice from T, E, and/or R represents how preference was distributed among total return, expected RoR, and risk respectively in the weighting; e.g., {T,R} indicates the scenario where total return and risk are valued equally and exclusively, therefore corresponding to the weighting $[a = 0.5,\ b = 0,\ c = 0.5]$.

Looking at the figure, we observe the following:

- In the {T} and {T,E} scenarios, the SOO solution dominates a handful of solutions of MOO3's Pareto front, but neither SOO nor the MOO3 designated solution dominates each other.



a) (a=1, b=0, c=0) / {T}

b) (a=0, b=1, c=0) / {E}

c) (a=0, b=0, c=1) / {R}

d) (a=0.5, b=0.5, c=0) / {T,E}

e) (a=0.5, b=0, c=0.5) / {T,R}

f) (a=0, b=0.5, c=0.5) / {E,R}

g) (a=0.33, b=0.33, c=0.33) / {T,E,R}

Figure 6.2: Representative run of the MOO3 and SOO algorithms for Apple stock. The SOO and MOO3-designated solutions are overlaid onto the MOO3 Pareto front for all seven scenarios considered. Risk, expected rate of return, and total return are shown here in their original units, rather than the normalised ones used during fitness evaluation.

- In the {E} scenario, SOO dominates the MOO3-designated solution, as well as the (unusually small) Pareto front.
- In the {R}, {T,R}, {E,R}, and {T,E,R} scenarios, the MOO3 designated solution dominates the SOO solution, and SOO does not dominate any solutions on the Pareto front.

The above suggests that, in this example at least, MOO3 exhibited superior performance when non-singular objectives were considered, showing its ability to consider multiple objectives naturally, driven by the concept of the Pareto front, as compared to an aggregate approach which imposes an arbitrary relationship between the objectives. By contrast, when dealing with single-objective scenarios, SOO was able to compete well with MOO3, in particular when considering single-objective scenarios, and SOO achieved a better total return and expected RoR in the {T} and {E} scenarios, respectively. However, MOO3 achieved better risk for the {R} scenario. This demonstrates that MOO3 still has the ability to generate good-quality single-objective designated solutions that may compete with explicitly single-objective approaches, even though the Pareto front-driven evolution process has to take into account and accommodate multiple objectives, partly acting as constraints. In the next section, we perform statistical analyses over multiple runs in order to examine how the two algorithms compare over multiple runs and datasets more generally.

6.4.2.3 Comparison of MOO3 to corresponding SOO approaches

Table 6.14 and Table6.15 present the summary statistics for SOO and MOO3 across four metrics, specifically total return (TR), expected rate of return ($\mathbb{E}[RoR]$), risk, and portfolio Sharpe ratio. Recall that the portfolio Sharpe ratio assumes an equal distribution of the 110 stocks (i.e., each stock has the same weight in the portfolio), taking only a single value in the table. For the remaining three metrics (TR, $\mathbb{E}[RoR]$, Risk) we present the mean, median, standard deviation, maximum and minimum values across the results from the 110 stocks. The left-hand side of the table presents the results for the single-objective (SOO) GP, while the right-hand side presents the results for the multi-objective MOO3 algorithm. Results are separated by the seven different sets as $[a, b, c]$ weights in Section 6.4.2.1. To enhance clarity, we also present the algorithms in the format $SOO_{TrRorRisk}$ (or $MOO_{TrRorRisk}$). The presence

of TR, $\mathbb{E}[RoR]$, and/or Risk in this naming convention depends on whether the [a,b,c] weight values are non-zero, where a corresponds to Tr, b to $\mathbb{E}[RoR]$, and c to Risk.

As we can observe, MOO3 generally has higher mean and median TR across different weight setups as compared to SOO across both 5 and 10-year periods. In addition, MOO3 shows a higher mean and median $\mathbb{E}[RoR]$ relative to SOO for most weight setups. With regard to risk, both algorithms show similar risk levels, but MOO3 offers small improvements in a few weight setups. Lastly, MOO3 has a higher Sharpe Ratio in 12 out of the 14 weights set-ups, suggesting better risk-adjusted portfolio returns.

We again perform a Kolmogorov-Smirnov (KS) test for each of the pairwise comparisons between SOO and MOO3 on three metrics, specifically the Total Return (TR), expected rate of return ($\mathbb{E}[RoR]$), and Risk. Recall that the null hypothesis for each KS test is that the two samples (SOO and MOO3) come from the same population distribution. Since we are conducting multiple comparisons, we again apply the Holm-Bonferroni correction to control the family-wise error rate at a significance level of 5%. Specifically, we adjust the p-values for the three comparisons (TR, $\mathbb{E}[RoR]$, Risk) within each $[a, b, c]$ weights set-up. The adjusted significance level for each rank is calculated in Equation 4.5.

Therefore, we compare the ranked p-values against these thresholds to determine if there is a significant difference arising between the SOO and MOO3 samples for each metric. The ranked p-values indicate a significant difference if the smallest p-value is less than 0.0167, the second smallest is less than 0.025, and the largest is less than 0.05.

Table 6.16 and Table 6.17 present the KS test results, along with the p-value and the adjusted significance level across the different weight set-ups ($[a, b, c]$). As we can observe, the differences between MOO3 and SOO are statistically significant at the 5% level in 9 out of 21 across the 5-year period and 11 out of 21 over the 10-year period. Particularly, MOO3 shows statistically significant improvements in TR and $\mathbb{E}[RoR]$ in most weight configurations and also in managing risk in some set-ups. The above findings lead us to conclude that our multi-objective optimisation GP algorithm provides a more robust and efficient optimisation strategy as compared to the single-objective optimization GP, particularly in terms of balancing return and risk.

SOO					MOO3				
[a, b, c]	**TR**	**$\mathbb{E}$[RoR]**	**Risk**	**SR**		**TR**	**$\mathbb{E}$[RoR]**	**Risk**	**SR**
[0.5, 0.5, 0] (SOO_{TrRor})					(MOO_{TrRor})				
Mean	17.71%	1.89%	0.10	0.60	**Mean**	**26.90%**	**2.90%**	0.10	**0.76**
Median	11.82%	1.45%	0.08		**Median**	**20.12%**	**2.32%**	0.08	
StDev	**0.35**	**0.03**	**0.07**		**StDev**	0.43	0.04	0.09	
Max	269.66%	23.85%	**0.52**		**Max**	**293.31%**	**24.35%**	0.61	
Min	**-56.45%**	**-5.84%**	0.01		**Min**	-59.28%	-11.59%	0.01	
[0, 0.5, 0.5] ($SOO_{RorRisk}$)					($MOO_{RorRisk}$)				
Mean	6.95%	0.91%	**0.07**	0.48	**Mean**	**15.52%**	**1.96%**	0.08	**0.78**
Median	8.23%	0.98%	0.06		**Median**	**13.52%**	**1.64%**	0.06	
StDev	**0.16**	0.02	**0.04**		**StDev**	0.22	0.02	0.06	
Max	66.63%	8.67%	**0.31**		**Max**	**133.75%**	**13.23%**	0.40	
Min	-74.83%	-8.70%	0.02		**Min**	**-33.96%**	**-3.24%**	**0.01**	
[0.5, 0, 0.5] (SOO_{TrRisk})					(MOO_{TrRisk})				
Mean	**22.48%**	1.82%	0.09	0.79	**Mean**	22.40%	**1.85%**	**0.07**	**0.89**
Median	16.65%	1.52%	0.08		**Median**	**17.45%**	**1.69%**	**0.06**	
StDev	0.38	0.02	0.06		**StDev**	**0.31**	0.02	0.06	
Max	**264.96%**	8.62%	**0.33**		**Max**	173.63%	**11.38%**	0.35	
Min	**-47.92%**	-5.34%	0.01		**Min**	-65.99%	**-5.18%**	0.01	
[0.33, 0.33, 0.33] ($SOO_{TrRorRisk}$)					($MOO_{TrRorRisk}$)				
Mean	20.98%	2.01%	0.09	0.77	**Mean**	**26.38%**	**2.51%**	0.09	**0.86**
Median	15.44%	1.73%	0.07		**Median**	**21.24%**	**2.27%**	0.07	
StDev	**0.36**	0.03	**0.06**		**StDev**	0.38	0.03	0.07	
Max	249.33%	10.64%	**0.39**		**Max**	**251.47%**	**13.20%**	0.53	
Min	**-51.36%**	**-5.06%**	0.01		**Min**	-54.43%	-6.58%	0.01	
[1, 0, 0] (SOO_{Tr})					(MOO_{Tr})				
Mean	25.99%	2.41%	0.09	0.76	**Mean**	**27.26%**	**2.93%**	0.09	**0.86**
Median	16.89%	1.93%	0.07		**Median**	**21.55%**	**2.33%**	**0.06**	
StDev	0.38	0.03	0.07		**StDev**	**0.33**	0.03	0.07	
Max	210.30%	17.80%	0.46		**Max**	**219.21%**	**21.90%**	**0.45**	
Min	**-47.43%**	**-3.52%**	0.01		**Min**	-48.60%	-6.90%	0.01	
[0, 1, 0] (SOO_{Ror})					(MOO_{Ror})				
Mean	16.26%	2.45%	0.13	0.44	**Mean**	**23.58%**	**3.04%**	**0.10**	**0.62**
Median	8.23%	1.42%	0.10		**Median**	**17.60%**	**2.28%**	**0.08**	
StDev	**0.38**	0.06	0.11		**StDev**	0.39	**0.05**	0.11	
Max	240.86%	**45.15%**	**0.90**		**Max**	**270.92%**	38.56%	1.01	
Min	**-34.53%**	**-4.28%**	0.02		**Min**	-36.42%	-11.19%	**0.01**	
[0, 0, 1] (SOO_{Risk})					(MOO_{Risk})				
Mean	2.70%	0.48%	0.08	0.18	**Mean**	**7.21%**	**0.94%**	**0.06**	**0.49**
Median	0.93%	0.22%	0.06		**Median**	**5.30%**	**0.79%**	**0.05**	
StDev	0.17	0.03	0.07		**StDev**	**0.15**	**0.02**	**0.05**	
Max	57.48%	**17.12%**	0.52		**Max**	**64.34%**	6.91%	**0.40**	
Min	**-62.32%**	**-6.77%**	0.02		**Min**	-63.21%	-9.26%	**0.01**	

Table 6.14: Summary statistics between SOO and MOO3 for a 5-year period. The best value for each respective metric is shown in boldface. The values of a, b, c correspond to the weights as defined in Section 6.4.2.1.

SOO					MOO3				
[a, b, c]	**TR**	**$\mathbb{E}$[RoR]**	**Risk**	**SR**		**TR**	**$\mathbb{E}$[RoR]**	**Risk**	**SR**
[0.5, 0.5, 0] (SOO_{TrRor})					(MOO_{TrRor})				
Mean	38.78%	2.08%	0.09	0.72	**Mean**	**51.90%**	**2.76%**	0.09	**1.10**
Median	24.94%	1.79%	0.07		**Median**	**37.88%**	**2.38%**	0.07	
StDev	**0.60**	0.03	0.07		**StDev**	0.61	**0.02**	**0.06**	
Max	**381.49%**	**21.66%**	**0.39**		**Max**	324.26%	13.07%	0.41	
Min	-63.09%	-5.17%	0.01		**Min**	**-42.72%**	**-2.98%**	**0.00**	
[0, 0.5, 0.5] ($SOO_{RorRisk}$)					($MOO_{RorRisk}$)				
Mean	11.94%	0.80%	0.06	0.57	**Mean**	**29.45%**	**1.85%**	0.06	**1.03**
Median	6.03%	0.69%	0.05		**Median**	**17.97%**	**1.64%**	0.05	
StDev	**0.28**	**0.01**	0.04		**StDev**	0.41	0.02	0.04	
Max	136.38%	4.72%	0.22		**Max**	**268.50%**	**10.59%**	**0.24**	
Min	-106.33%	-4.42%	0.01		**Min**	**-27.55%**	**-2.22%**	**0.00**	
[0.5, 0, 0.5] (SOO_{TrRisk})					(MOO_{TrRisk})				
Mean	40.11%	**2.03%**	0.08	0.86	**Mean**	**41.58%**	1.63%	**0.06**	**1.12**
Median	**26.89%**	**1.53%**	0.06		**Median**	24.18%	1.44%	**0.05**	
StDev	0.62	0.02	0.06		**StDev**	**0.52**	**0.01**	**0.04**	
Max	**314.31%**	**11.20%**	0.33		**Max**	258.04%	6.80%	**0.29**	
Min	-219.85%	-6.47%	0.01		**Min**	**-37.47%**	**-1.23%**	0.01	
[0.33, 0.33, 0.33] ($SOO_{TrRorRisk}$)					($MOO_{TrRorRisk}$)				
Mean	44.73%	2.17%	0.08	0.84	**Mean**	**50.47%**	**2.51%**	0.08	**1.03**
Median	31.18%	1.66%	0.07		**Median**	**34.98%**	**2.16%**	**0.06**	
StDev	**0.57**	0.03	0.05		**StDev**	0.61	**0.02**	0.05	
Max	**312.83%**	**16.18%**	0.34		**Max**	285.05%	11.42%	0.34	
Min	-43.55%	-6.52%	0.01		**Min**	**-38.68%**	**-3.74%**	0.01	
[1, 0, 0] (SOO_{Tr})					(MOO_{Tr})				
Mean	47.36%	2.23%	0.08	**0.79**	**Mean**	**61.33%**	**3.28%**	0.08	0.66
Median	31.25%	1.66%	0.06		**Median**	**41.68%**	**2.52%**	0.06	
StDev	**0.64**	**0.03**	0.06		**StDev**	0.67	0.05	0.06	
Max	**366.13%**	18.63%	**0.39**		**Max**	341.71%	**44.36%**	0.43	
Min	-46.60%	-4.05%	0.02		**Min**	**-44.63%**	**-4.03%**	**0.01**	
[0, 1, 0] (SOO_{Ror})					(MOO_{Ror})				
Mean	29.46%	2.08%	0.11	0.48	**Mean**	**48.29%**	**3.38%**	**0.09**	**0.76**
Median	15.92%	1.50%	0.08		**Median**	**32.81%**	**2.62%**	**0.07**	
StDev	0.71	0.04	0.08		**StDev**	**0.69**	0.04	**0.07**	
Max	**552.10%**	24.58%	0.52		**Max**	544.98%	**25.92%**	**0.49**	
Min	-63.57%	-6.10%	0.01		**Min**	**-44.89%**	**-4.28%**	0.01	
[0, 0, 1] (SOO_{Risk})					(MOO_{Risk})				
Mean	3.21%	0.14%	0.06	0.10	**Mean**	**15.15%**	**0.99%**	**0.05**	**0.65**
Median	-0.59%	0.14%	0.05		**Median**	**10.06%**	**0.86%**	**0.03**	
StDev	0.32	**0.01**	**0.04**		**StDev**	**0.24**	0.02	0.05	
Max	**279.63%**	5.62%	**0.19**		**Max**	152.06%	**12.24%**	0.42	
Min	-45.49%	-4.92%	0.01		**Min**	**-32.11%**	**-1.98%**	0.01	

Table 6.15: Summary statistics between SOO and MOO3 in a 10-year period. The best value for each respective metric is shown in boldface. The values of a, b, c correspond to the weights as defined in Section 6.4.2.1.

Table 6.16: Kolmogorov-Smirnov test results p-values for different weight set-ups ($[a, b, c]$) for a 5-year period. Statistically significant results at the 5% level are denoted in boldface.

[a, b, c]	**Metric**	**p-value**	**Adjusted significance level**
[0.5, 0.5, 0]	TR	**0.0137**	0.025 (0.05/2)
	$\mathbb{E}[RoR]$	**0.0088**	0.0167 (0.05/3)
	Risk	0.6186	0.05 (0.05/1)
[0, 0.5, 0.5]	TR	**2.33E-04**	0.025 (0.05/2)
	$\mathbb{E}[RoR]$	**2.46E-06**	0.0167 (0.05/3)
	Risk	0.8399	0.05 (0.05/1)
[0.5, 0, 0.5]	TR	0.506	0.025 (0.05/2)
	$\mathbb{E}[RoR]$	0.9237	0.05 (0.05/1)
	Risk	**1.29E-04**	0.0167 (0.05/3)
[0.33, 0.33, 0.33]	TR	0.1312	0.05 (0.05/1)
	$\mathbb{E}[RoR]$	0.0944	0.0167 (0.05/3)
	Risk	0.0944	0.025 (0.05/2)
[1, 0, 0]	TR	0.4033	0.0167 (0.05/3)
	$\mathbb{E}[RoR]$	0.5060	0.025 (0.05/2)
	Risk	0.8399	0.05 (0.05/1)
[0, 1, 0]	TR	**0.0034**	0.025 (0.05/2)
	$\mathbb{E}[RoR]$	**0.0461**	0.05 (0.05/1)
	Risk	**0.0034**	0.0167 (0.05/3)
[0, 0, 1]	TR	**0.0012**	0.025 (0.05/2)
	$\mathbb{E}[RoR]$	**2.33E-04**	0.0167 (0.05/3)
	Risk	**0.0209**	0.05 (0.05/1)

While the above KS test was useful for making pairwise comparisons between the SOO and MOO3 algorithms under different weight set-ups and metrics, we are also interested in gaining a better understanding of the overall performance of all SOO and MOO3 algorithms across the different weight set-ups. The non-parametric Friedman test is applied to evaluate the comparison among multiple algorithms across multiple datasets. In each Friedman test (one per TR, $\mathbb{E}[RoR]$, and Risk metric), the algorithms included are the SOO and MOO3 algorithms under the different weight set-ups.

Additionally, we applied the Hommel post-hoc test to ascertain the significance of the differences between the average ranks. We present both in Table 6.18 and Table6.19. For each algorithm, the table shows the average rank (first column), and the adjusted $p-$value

Table 6.17: Kolmogorov-Smirnov test results p-values for different weight set-ups ($[a, b, c]$) across a 10-year period. Statistically significant results at the 5% level are denoted in boldface.

[a, b, c]	**Metric**	**p-value**	**Adjusted significance level**
[0.5, 0.5, 0]	TR	0.3141	0.025 (0.05/2)
	$\mathbb{E}[RoR]$	0.0209	0.0167 (0.05/3)
	Risk	0.9751	0.05 (0.05/1)
[0, 0.5, 0.5]	TR	**3.71E-05**	0.0167 (0.05/3)
	$\mathbb{E}[RoR]$	**6.98E-05**	0.025 (0.05/2)
	Risk	0.0666	0.05 (0.05/1)
[0.5, 0, 0.5]	TR	0.506	0.05 (0.05/1)
	$\mathbb{E}[RoR]$	0.179	0.025 (0.05/2)
	Risk	**0.0088**	0.0167 (0.05/3)
[0.33, 0.33, 0.33]	TR	0.7336	0.05 (0.05/1)
	$\mathbb{E}[RoR]$	0.0944	0.0167 (0.05/3)
	Risk	0.2395	0.025 (0.05/2)
[1, 0, 0]	TR	0.3141	0.025 (0.05/2)
	$\mathbb{E}[RoR]$	**0.0137**	0.0167 (0.05/3)
	Risk	0.9958	0.05 (0.05/1)
[0, 1, 0]	TR	**0.0034**	0.025 (0.05/2)
	$\mathbb{E}[RoR]$	**0.0021**	0.0167 (0.05/3)
	Risk	0.0313	0.05 (0.05/1)
[0, 0, 1]	TR	**5.41E-08**	0.025 (0.05/2)
	$\mathbb{E}[RoR]$	**2.38E-08**	0.0167 (0.05/3)
	Risk	**2.33E-04**	0.05 (0.05/1)

of the statistical test when that algorithm's average rank is compared to the average rank of the algorithm with the best rank (control algorithm) according to Hommel's post-hoc test (second column). When statistically significant differences arise between the average ranks of an algorithm and the control algorithm at the 5% level ($p \leq 0.05$), the relevant average rank is put in boldface.

As we can observe from the total return results of Table 6.19, the control algorithm is MOO_{Tr}, i.e. the MOO GP algorithm that extracts the best model from the Pareto front based on the highest total return value. In addition, it statistically outperforms all other algorithms significantly at the 5% level. Given that we are currently considering total re-

(a) Total Return

	Avg Rank	Adj p_{Homm}
MOO$_{Tr}$	4.936364	-
MOO$_{TrRorRisk}$	5.418182	1.44E-01
MOO$_{TrRor}$	5.554545	1.06E-01
SOO$_{Tr}$	**5.927273**	3.51E-02
MOO$_{Ror}$	**5.936364**	3.32E-02
SOO$_{TrRisk}$	**6.690909**	1.20E-03
MOO$_{TrRisk}$	**6.845455**	4.69E-04
SOO$_{TrRorRisk}$	**6.954545**	2.66E-04
MOO$_{RorRisk}$	**7.863636**	2.50E-07
SOO$_{TrRor}$	**8.036364**	5.81E-08
SOO$_{Ror}$	**8.627273**	1.55E-10
SOO$_{RorRisk}$	**10.209091**	1.94E-19
MOO$_{Risk}$	**10.372727**	1.95E-20
SOO$_{Risk}$	**11.545455**	9.87E-29

(b) Expected Rate of Return

	Avg Rank	Adj p_{Homm}
MOO$_{TrRor}$	4.945455	-
MOO$_{Tr}$	5.136364	2.25E-01
MOO$_{Ror}$	5.290909	1.75E-01
MOO$_{TrRorRisk}$	**5.809091**	4.97E-02
SOO$_{Tr}$	**6.645455**	1.27E-03
MOO$_{RorRisk}$	**7.118182**	7.12E-05
SOO$_{TrRorRisk}$	**7.354545**	1.58E-05
SOO$_{Ror}$	**7.390909**	1.22E-05
SOO$_{TrRisk}$	**7.754545**	7.81E-07
MOO$_{TrRisk}$	**7.863636**	2.76E-07
SOO$_{TrRor}$	**8.063636**	4.82E-08
SOO$_{RorRisk}$	**10.109091**	9.66E-19
MOO$_{Risk}$	**10.109091**	9.66E-19
SOO$_{Risk}$	**11.309091**	7.37E-27

(c) Risk

	Avg Rank	Adj p_{Homm}
MOO$_{Risk}$	2.754545	-
MOO$_{TrRisk}$	**4.490909**	8.21E-04
SOO$_{RorRisk}$	**5.463636**	8.90E-07
MOO$_{RorRisk}$	**5.536364**	5.34E-07
SOO$_{Risk}$	**6.063636**	4.17E-09
MOO$_{TrRorRisk}$	**7.063636**	5.99E-14
MOO$_{Tr}$	**7.754545**	4.06E-18
SOO$_{Tr}$	**8.072727**	4.43E-20
SOO$_{TrRisk}$	**8.427273**	1.54E-22
SOO$_{TrRorRisk}$	**8.800000**	3.21E-25
MOO$_{Ror}$	**9.063636**	2.76E-27
MOO$_{TrRor}$	**9.663636**	5.50E-32
SOO$_{TrRor}$	**9.763636**	1.22E-32
SOO$_{Ror}$	**12.036364**	3.28E-54

Table 6.18: Statistical test results for average TR, $\mathbb{E}[RoR]$, Risk, and Fitness function, according to the non-parametric Friedman test with the Hommel post-hoc test of different MOO and SOO algorithms across the 5-year period. The subscript for each algorithm denotes which metrics were optimised. When more than one metric is present, equal weights have been assigned to each metric. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface, indicating that the adjusted p-value is less than 0.05.

(a) Total Return

	Avg Rank	Adj p_{Homm}
MOO$_{Tr}$	4.018182	-
MOO$_{TrRor}$	**5.600000**	2.58E-03
MOO$_{TrRorRisk}$	**5.863636**	6.34E-04
SOO$_{TrRorRisk}$	**5.909091**	4.95E-04
SOO$_{Tr}$	**6.027273**	2.21E-04
SOO$_{TrRisk}$	**6.136364**	1.07E-04
MOO$_{Ror}$	**6.472727**	1.06E-05
SOO$_{TrRor}$	**6.718182**	1.51E-06
MOO$_{TrRisk}$	**7.145455**	3.28E-08
MOO$_{RorRisk}$	**8.318182**	5.77E-14
SOO$_{Ror}$	**9.300000**	9.98E-20
MOO$_{Risk}$	**10.454545**	8.41E-28
SOO$_{RorRisk}$	**11.009091**	2.98E-32
SOO$_{Risk}$	**11.945455**	1.95E-40

(b) Expected Rate of Return

	Avg Rank	Adj p_{Homm}
MOO$_{Ror}$	4.627273	-
MOO$_{TrRor}$	4.736364	2.34E-01
MOO$_{Tr}$	4.809091	2.17E-01
MOO$_{TrRorRisk}$	**5.700000**	3.42E-05
SOO$_{Tr}$	**6.763636**	8.05E-05
SOO$_{TrRorRisk}$	**6.890909**	3.42E-05
SOO$_{TrRisk}$	**7.045455**	1.18E-05
SOO$_{TrRor}$	**7.163636**	5.09E-06
MOO$_{RorRisk}$	**7.436364**	5.21E-07
SOO$_{Ror}$	**8.000000**	2.91E-09
MOO$_{TrRisk}$	**8.318182**	1.06E-10
MOO$_{Risk}$	**10.390909**	4.74E-23
SOO$_{RorRisk}$	**10.781818**	8.44E-26
SOO$_{Risk}$	**12.209091**	3.31E-37

(c) Risk

	Avg Rank	Adj p_{Homm}
MOO$_{Risk}$	2.163636	-
MOO$_{TrRisk}$	**4.845455**	1.01E-06
SOO$_{Risk}$	**5.109091**	1.04E-07
SOO$_{RorRisk}$	**5.163636**	6.55E-08
MOO$_{RorRisk}$	**5.300000**	1.66E-08
MOO$_{TrRorRisk}$	**7.509091**	2.10E-20
MOO$_{Tr}$	**7.681818**	1.42E-21
SOO$_{TrRisk}$	**7.990909**	8.88E-24
SOO$_{Tr}$	**8.300000**	4.43E-26
SOO$_{TrRorRisk}$	**9.263636**	1.03E-33
MOO$_{Ror}$	**9.709091**	1.35E-37
SOO$_{TrRor}$	**10.027273**	1.82E-40
MOO$_{TrRor}$	**10.145455**	1.69E-41
SOO$_{Ror}$	**11.754545**	1.52E-57

Table 6.19: Statistical test results for average TR, $\mathbb{E}[RoR]$, Risk, and Fitness function, according to the non-parametric Friedman test with the Hommel post-hoc test of different MOO and SOO algorithms over the 10-year period. The subscript for each algorithm denotes which metrics were optimised. When more than one metric is present, equal weights have been assigned to each metric. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface, indicating that the adjusted p-value is lower than 0.05.

turn results in this subtable, it is unsurprising that the control algorithm prioritises total return. It is also worth noting that the second and third-ranked algorithms are also MOO-based, namely MOO_{TrRor} and $MOO_{TrRorRisk}$. The former gives equal weight to total return and risk ($a = b = 0.5, c = 0$), while the latter assigns equal weight to all three metrics ($a = b = c = 0.33$). This is an important finding because it demonstrates that these multi-objective algorithms are able to outrank all SOO algorithms, thereby confirming the fact that considering multiple objectives under a MOO framework like NSGA-II has significant benefits over the SOO algorithm that employs an aggregate fitness function. Similar findings can be observed for the expected rate of return and risk results (Tables 6.19b, 6.18b, 6.19c, and 6.18c). More specifically, when looking into the expected rate of return results over the 10-year period, the best rank algorithm is MOO_{Ror}, which optimises only the expected RoR. In the next three positions, there are again MOO algorithms that optimise different combinations of the metrics, even risk ($MOO_{TrRorRisk}$. This again shows the advantages of MOO. Further, when looking into the risk results, MOO_{Risk} ranks first, followed by MOO_{TrRisk}. So we again see the same pattern, whereby the algorithm is specifically optimised for a particular metric (risk in this instance) and tends to perform best in this specific metric, followed by another MOO algorithm that considers two or three objectives. The only difference is that, in Table 6.18a, the top-ranked algorithm is MOO_{TrRor} rather than MOO_{Ror}. However, considering the p-value of MOO_{Ror} and MOO_{Tr} are above 5%, no significant difference arises, and the finding remains unchanged. Finally, it is also worth noting that, in the top 2 positions of each suitable, we have a MOO_{Tr} variant (i.e. a MOO3 algorithm that has used total return [alone or in consideration with another metric] to extract the best model from the Pareto front). This suggests that prioritising total return in the optimisation process can lead to favourable outcomes.

To conclude, the Friedman test results highlight the effectiveness of the MOO3 algorithms over their SOO counterparts. Combining up to two metrics tends to be beneficial (or at least have comparable performance to the MOO algorithms optimising a single metric), as we have seen in the case of MOO_{TrRor}, which performs well for both total return and expected RoR. However, the trade-offs become apparent when more metrics are combined (e.g.

$MOO_{TrRorRisk}$), where balancing multiple objectives leads to a slight drop in the performance of the individual metrics.

6.4.2.4 Comparison of MOO3 algorithms to other technical indicators

The previous section of this chapter has established that our proposed multi-objective GP algorithm can outperform its SOO counterpart. In the current section, we are interested in benchmarking the MOO3 algorithm to MACD, OBV, and MTM, as Section 6.4.2.4. From Table 6.18 and Table 6.19, we concluded that MOO_{Tr}, MOO_{Ror}, and MOO_{Risk} rank first in their corresponding Friedman tests. We hence bring forward these algorithms to be compared against the technical indicators. In addition to MOO3 algorithms optimising a single metric, we are also interested in bringing forward comparison algorithms that optimise more than one metric and thus we will also compare MOO_{TrRor} and MOO_{TrRisk} against the technical analysis benchmarks.

Table 6.20 and Table 6.21 present the results across TR, $\mathbb{E}[RoR]$, risk, and portfolio Sharpe ratio. As we can observe, in all cases, the MOO3 algorithms have significantly improved values over MACD, OBV, and MTM. As previously noted, MOO_{Tr} and MOO_{Ror} offer the best performance for total return and expected RoR, respectively, and generally have a good overall performance in maximising returns. However, this comes at the cost of higher risk. On the other hand, MOO_{Risk} minimises risk effectively, but at the cost of lower returns. Lastly, $MOOTrRisk$ offers a balanced approach with a high Sharpe ratio, indicating good risk-adjusted performance.

The above findings are further supported by the Friedman tests (one per metric) presented in Table 6.22 and Table 6.23. In the half tables, we can observe that, statistically, the first rank MOO3 algorithm significantly outperforms all other algorithms at the 5% significance level. In addition, the majority of the MOO algorithms outrank the trading strategies derived by using technical indicators.

Table 6.20: Summary statistics of the three-objective optimisation algorithms and technical indicators over the 5-year period. We use boldface for the best values for each measure.

Measure	Algorithm							
	Total return							
	MOO_{Tr}	MOO_{Ror}	MOO_{Risk}	MOO_{TrRor}	MOO_{TrRisk}	MACD	OBV	MTM
Average	**27.26%**	23.58%	7.21%	26.90%	22.40%	7.80%	2.93%	6.72%
Median	**21.55%**	17.60%	5.30%	20.12%	17.45%	0.01%	0.01%	0.48%
Standard deviation	0.33	0.39	**0.15**	0.43	0.31	0.39	0.31	0.48
Max	219.21%	270.92%	64.34%	**293.31**%	173.63%	191.07%	152.31%	252.25%
Min	-48.60%	**-36.42**%	-63.21%	-59.28%	-65.99%	-97.24%	-98.39%	-98.49%
	Expected rate of return							
	MOO_{Tr}	MOO_{Ror}	MOO_{Risk}	MOO_{TrRor}	MOO_{TrRisk}	MACD	OBV	MTM
Average	2.93%	**3.04%**	0.94%	2.90%	1.85%	1.31%	-0.03%	0.32%
Median	**2.33**%	2.28%	0.79%	2.32%	1.69%	0.00%	0.00%	0.02%
Standard deviation	0.03	0.05	**0.02**	0.04	**0.02**	0.07	0.05	**0.02**
Max	32.90%	**38.56**%	6.91%	24.35%	11.38%	31.84%	21.52%	10.98%
Min	-6.90%	-11.19%	-9.26%	-11.59%	-5.18%	-24.31%	-24.60%	**-2.59**%
	Risk							
	MOO_{Tr}	MOO_{Ror}	MOO_{Risk}	MOO_{TrRor}	MOO_{TrRisk}	MACD	OBV	MTM
Average	0.09	0.10	**0.06**	0.10	0.07	0.09	0.07	0.12
Median	0.06	0.08	**0.05**	0.08	0.06	**0.05**	**0.05**	0.08
Standard deviation	0.07	0.11	**0.05**	0.09	0.06	0.15	0.07	0.15
Max	0.45	1.01	0.40	0.61	**0.35**	1.05	**0.35**	0.95
Min	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02
	Sharpe ratio							
	MOO_{Tr}	MOO_{Ror}	MOO_{Risk}	MOO_{TrRor}	MOO_{TrRisk}	MACD	OBV	MTM
SR	0.86	0.62	0.49	0.76	**0.89**	0.19	-0.01	0.17

Table 6.21: Summary statistics of the three-objective optimisation algorithms and technical indicators over the 10-year period. We use boldface for the best values for each measure.

Measure	Algorithm							
	Total return							
	MOO_{Tr}	MOO_{Ror}	MOO_{Risk}	MOO_{TrRor}	MOO_{TrRisk}	MACD	OBV	MTM
Average	**61.33%**	48.29%	15.15%	51.90%	41.58%	13.99%	2.63%	18.43%
Median	**41.68%**	32.81%	10.06%	37.88%	24.18%	2.22%	1.73%	7.44%
Standard deviation	0.67	0.69	**0.24**	0.61	0.52	0.58	0.43	0.67
Max	341.71%	**544.98%**	152.06%	324.26%	258.04%	360.11%	148.28%	514.56%
Min	-44.63%	-44.89%	**-32.11%**	-42.72%	-37.47%	-90.38%	-177.72%	-112.67%
	Expected rate of return							
	MOO_{Tr}	MOO_{Ror}	MOO_{Risk}	MOO_{TrRor}	MOO_{TrRisk}	MACD	OBV	MTM
Average	3.28%	**3.38%**	0.99%	2.76%	1.63%	1.71%	0.77%	0.32%
Median	2.52%	**2.62%**	0.86%	2.38%	1.44%	0.14%	0.18%	0.11%
Standard deviation	0.05	0.04	0.02	0.02	**0.01**	0.06	0.05	**0.01**
Max	**44.36%**	25.92%	12.24%	13.07%	6.80%	36.01%	35.65%	6.95%
Min	-4.03%	-4.28%	-1.98%	-2.98%	**-1.23%**	-9.04%	-11.11%	-1.48%
	Risk							
	MOO_{Tr}	MOO_{Ror}	MOO_{Risk}	MOO_{TrRor}	MOO_{TrRisk}	MACD	OBV	MTM
Average	0.08	0.09	**0.05**	0.09	0.06	0.12	0.12	0.15
Median	0.06	0.07	**0.03**	0.07	0.05	0.07	0.09	0.09
Standard deviation	0.06	0.07	0.05	0.06	**0.04**	0.22	0.12	0.20
Max	0.43	0.49	0.42	0.41	**0.29**	1.58	0.76	1.82
Min	0.01	0.01	0.01	0.01	0.01	**0.00**	**0.00**	0.03
	Sharpe ratio							
	MOO_{Tr}	MOO_{Ror}	MOO_{Risk}	MOO_{TrRor}	MOO_{TrRisk}	MACD	OBV	MTM
SR	0.66	0.76	0.65	1.10	**1.12**	0.24	0.06	0.28

Table 6.22: Statistical test results between MOO3 algorithms and technical analysis indicators for average TR, $\mathbb{E}[RoR]$, Risk, and Fitness function, according to the non-parametric Friedman test with the Hommel post-hoc test of different MOO and SOO algorithms over the 5-year period. The subscript for each algorithm denotes which metrics were optimised. When more than one metric is present, equal weights have been assigned to each metric. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface indicating that the adjusted p-value is lower than 0.05.

(a) Total Return

	Avg Rank	Adj p_{Homm}
MOO_{Tr}(c)	2.900000	-
MOO_{TrRor}	3.136364	1.98E-01
MOO_{Ror}	3.500000	3.71E-02
MOO_{TrRisk}	**3.972727**	7.36E-04
MACD	**5.445455**	9.53E-14
MOO_{Risk}	**5.545455**	2.07E-14
MTM	**5.709091**	5.73E-16
OBV	**5.736364**	5.73E-16

(b) Expected Rate of Return

	Avg Rank	Adj p_{Homm}
MOO_{TrRor}(c)	2.900000	-
MOO_{Tr}	2.981818	2.03E-01
MOO_{Ror}	3.109091	1.49E-01
MOO_{TrRisk}	**4.163636**	6.12E-05
MOO_{Risk}	**5.290909**	9.06E-13
MACD	**5.336364**	3.62E-13
OBV	**5.900000**	1.78E-18
MTM	**6.090909**	3.51E-20

(c) Risk

	Avg Rank	Adj p_{Homm}
MOO_{Risk}(c)	2.318182	-
MOO_{TrRisk}	**3.318182**	2.61E-03
MOO_{Tr}	**4.600000**	1.32E-11
MACD	**4.672727**	5.72E-12
MOO_{Ror}	**5.000000**	2.77E-15
MOO_{TrRor}	**5.245455**	2.25E-17
OBV	**5.309091**	6.98E-18
MTM	**5.427273**	6.95E-19

Table 6.23: Statistical test results between MOO3 algorithms and technical analysis indicators for average TR, $\mathbb{E}[RoR]$, Risk, and Fitness function, according to the non-parametric Friedman test with the Hommel post-hoc test of different MOO and SOO algorithms over the 10-year period. The subscript for each algorithm denotes which metrics were optimised. When more than one metric is present, equal weights have been assigned to each metric. Significant differences arising between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface, indicating that the adjusted p-value is lower than 0.05.

(a) Total Return

	Avg Rank	Adj p_{Homm}
MOO_{Tr}(c)	2.200000	-
MOO_{TrRor}	**3.072727**	3.62E-03
MOO_{Ror}	**3.527273**	3.00E-05
MOO_{TrRisk}	**3.863636**	2.99E-07
MOO_{Risk}	**5.600000**	6.08E-23
MTM	**5.636364**	2.97E-23
MACD	**5.827273**	2.03E-25
OBV	**6.272727**	6.32E-31

(b) Expected Rate of Return

	Avg Rank	Adj p_{Homm}
MOO_{Ror}(c)	2.809091	-
MOO_{TrRor}	2.881818	2.07E-01
MOO_{Tr}	3.018182	1.53E-01
MOO_{TrRisk}	**4.400000**	9.41E-07
MACD	**5.390909**	1.65E-14
MOO_{Risk}	**5.527273**	1.13E-15
OBV	**5.727273**	1.46E-17
MTM	**6.245455**	7.81E-23

(c) Risk

	Avg Rank	Adj p_{Homm}
MOO_{Risk}(c)	1.509091	-
MOO_{TrRisk}	**3.063636**	1.37E-06
MOO_{Tr}	**4.154545**	4.37E-15
MACD	**4.672727**	1.70E-20
MOO_{Ror}	**5.136364**	7.60E-26
MOO_{TrRor}	**5.218182**	9.42E-27
OBV	**5.645455**	3.23E-32
MTM	**6.600000**	7.28E-46

Table 6.24: Summary statistics for the three-objective optimisation algorithms and buy-and-hold in terms of total return over the 5-year period. The best values per metric appear in boldface.

Algorithm	MOO$_{Tr}$	MOO$_{Ror}$	MOO$_{Risk}$	MOO$_{TrRor}$	MOO$_{TrRisk}$	Buy-and-hold
Average	**27.26%**	23.58%	7.21%	26.90%	22.40%	13.31%
Median	**21.55%**	17.60%	5.30%	20.12%	17.45%	2.05%
Standard deviation	0.33	0.39	**0.15**	0.43	0.31	0.84
Max	219.21%	270.92%	64.34%	293.31%	173.63%	**755.68%**
Min	-48.60%	-36.42%	-63.21%	-59.28%	-65.99%	**-82.85%**

6.4.2.5 Buy-and-hold

We now compare the proposed MOO3 algorithms with the buy-and-hold strategy. Recall that, due to buy-and-hold making only a single trade (while the MOO GPs make several), it is fairer to compare them across the TR over the test set period rather than in terms of $\mathbb{E}[RoR]$ and risk.

Table 6.24 and Table 6.26 present the performance metrics of the MOO3 algorithms alongside the buy-and-hold strategy in the 5 and 10-year period. We can observe that buy-and-hold has a strong average performance (mainly due to outliers), while its median value of 11.44% in the 10-year period is only able to outperform MOO_{Risk}, which is understandable, given that the latter does not optimise total return. Further, the buy-and-hold strategy has the worst median value of 2.05% across the 5-year period. All other algorithms yield significantly higher average and median values. These results are also confirmed by the Friedman test presented in Table 6.25 and Table 6.27, which show that buy-and-hold is statistically and significantly outperformed by MOO_{Tr}.

6.4.3 Comparison of MOO2 algorithms to MOO3 algorithms

In the previous sections, both the αMOO2 algorithms and MOO3 algorithms showed a significant improvement as compared to the SOO algorithms, technical indicators, and buy-and-hold strategy. Now, we aim to compare these algorithms among themselves to select the most profitable and least risky trading strategy. To achieve this, we use the best performers from the αMOO2 algorithms, $\alpha MOO2^{RoR}_{Cos}$ and $\alpha MOO2^{RoR}_{Sig}$, and the MOO3 algorithms, MOO_{Tr}, MOO_{Ror}, MOO_{Risk}, MOO_{TrRor} and MOO_{TrRisk}.

Table 6.25: Statistical test results between MOO3 algorithms and technical analysis indicators for average TR, $\mathbb{E}[RoR]$, Risk, and Fitness function, according to the non-parametric Friedman test with the Hommel post-hoc test of different MOO and SOO algorithms. The subscript for each algorithm denotes which metrics were optimised. When more than one metric is present, equal weights have been assigned to each metric. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface, indicating that the adjusted p-value is lower than 0.05.

Algorithm	Avg rank	Adj p_{Homm}
MOO_{Tr} (c)	2.500000	-
MOO_{TrRor}	2.718182	1.04E-01
MOO_{Ror}	**3.127273**	4.87E-03
MOO_{TrRisk}	**3.445455**	9.03E-05
Buy-and-hold	**4.400000**	3.10E-13
MOO_{Risk}	**4.763636**	2.84E-17

Table 6.26: Summary statistics for the three-objective optimisation algorithms and buy-and-hold in terms of total return over the 10-year period. The best values per metric appear in boldface.

Algorithm	MOO_{Tr}	MOO_{Ror}	MOO_{Risk}	MOO_{TrRor}	MOO_{TrRisk}	Buy-and-hold
Average	**61.33%**	48.29%	15.15%	51.90%	41.58%	41.11%
Median	**41.68%**	32.81%	10.06%	37.88%	24.18%	11.44%
Standard deviation	0.67	0.69	**0.24**	0.61	0.52	1.81
Max	341.71%	544.98%	152.06%	324.26%	258.04%	**1753.05%**
Min	-44.63%	-44.89%	-32.11%	-42.72%	-37.47%	**-89.62%**

Table 6.27: Statistical test results between MOO3 algorithms and technical analysis indicators for average TR, $\mathbb{E}[RoR]$, Risk, and Fitness function, according to the non-parametric Friedman test with the Hommel post-hoc test of different MOO and SOO algorithms. The subscript for each algorithm denotes which metrics were optimised. When more than one metric is present, equal weights have been assigned to each metric. Significant differences between the control algorithm (denoted with (c) and the algorithms represented by a row at the $\alpha = 5\%$ level are shown in boldface, indicating that the adjusted p-value is lower than 0.05.

Algorithm	Avg rank	Adj p_{Homm}
MOO_{Tr} (c)	1.97	-
MOO_{TrRor}	**2.83**	1.24E-04
MOO_{Ror}	**3.26**	1.14E-07
MOO_{TrRisk}	**3.51**	6.24E-10
Buy-and-hold	**4.40**	2.88E-20
MOO_{Risk}	**5.00**	2.44E-20

From Table 6.28 and Table 6.29, we can observe that MOO3 algorithms exhibit better performance across all metrics including TR, $\mathbb{E}[RoR]$, risk, and the portfolio Sharpe ratio across both the 5- and 10-year datasets. The MOO2 algorithms only perform best in a few cases. For example, the best standard deviation of the $\mathbb{E}[RoR]$ and risk is achieved by the MOO2 algorithms. It could be explained by MOO3 algorithms taking advantage of optimising TR, while MOO2 algorithms do not. However, for the rest of the metrics, MOO3 still performs better results. MOO_{Ror} and MOO_{Tr} have the top two average and median $\mathbb{E}[RoR]$, which is almost 50% higher than the MOO2 algorithms. In addition, MOO_{Risk} performs best in terms of risk in both periods. Moreover, MOO_{TrRisk} has the best portfolio Sharpe ratio value.

To validate the above findings and identify the best model, we ran the Friedman non-parametric test in Table 6.30 and Table 6.34. The results indicate that MOO_{Tr} ranks first in terms of TR, significantly outperforming the other algorithms, except for MOO_{TrRor} for the 5-year data. Regarding $\mathbb{E}[RoR]$, MOO_{TrRor} performs best using the 5-year data, while MOO_{Ror} ranks first with the 10-year data. However, there is no significant difference between MOO_{TrRor}, MOO_{Ror}, and MOO_{Tr}. Additionally, the lowest risk values are achieved by MOO_{Risk}, which outperforms all other algorithms. In conclusion, it is evident that MOO3 algorithms statistically outperform MOO2 across all three metrics. Among the MOO3 algorithms, MOO_{Tr} performs the best, as it ranks in the top two for both TR and $\mathbb{E}[RoR]$, while maintaining the same distribution with MOO_{TrRor} for the 5-year data.

6.5 Conclusion

To conclude, this chapter presented two novel multi-objective genetic programming algorithms, MOO2 and MOO3, leveraging the NSGA-II algorithm. These algorithms incorporate features from both the event-based concept of directional changes and physical time. We used the α-dominance strategy to address the challenges with the convergence ability of the multi-objective GP in MOO2. Additionally, we defined the 'best' solutions from the Pareto front to satisfy the needs of traders by evaluating the performance of the training dataset. We compared the proposed MOO algorithms to the single-objective optimisation (SOO) method using

Table 6.28: Comparison between MOO2 and MOO3 algorithms on 5-year periods. Best values per row appear in boldface.

Measurement	Total return						
Algorithm	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{RoR}$	MOO_{Tr}	MOO_{Ror}	MOO_{Risk}	MOO_{TrRor}	MOO_{TrRisk}
Average	15.81%	14.78%	**27.26%**	23.58%	7.21%	26.90%	22.40%
Median	13.72%	11.76%	**21.55%**	17.60%	5.30%	20.12%	17.45%
Standard deviation	0.23	0.28	0.33	0.39	**0.15**	0.43	0.31
Max	155.12%	181.55%	219.21%	270.92%	64.34%	**293.31**%	173.63%
Min	-38.93%	-64.20%	-48.60%	**-36.42**%	-63.21%	-59.28%	-65.99%
Measurement	Expected Rate of Return						
Algorithm	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{RoR}$	MOO_{Tr}	MOO_{Ror}	MOO_{Risk}	MOO_{TrRor}	MOO_{TrRisk}
Average	1.82%	1.58%	2.93%	**3.04%**	0.94%	2.90%	1.85%
Median	1.66%	1.37%	**2.33**%	2.28%	0.79%	2.32%	1.69%
Standard deviation	**0.02**	**0.02**	0.03	0.05	**0.02**	0.04	**0.02**
Max	12.06%	11.24%	32.90%	**38.56**%	6.91%	24.35%	11.38%
Min	**-4.32**%	-8.10%	-6.90%	-11.19%	-9.26%	-11.59%	-5.18%
Measurement	Risk						
Algorithm	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{RoR}$	MOO_{Tr}	MOO_{Ror}	MOO_{Risk}	MOO_{TrRor}	MOO_{TrRisk}
Average	0.08	0.07	0.09	0.10	**0.06**	0.10	0.07
Median	0.06	0.06	0.06	0.08	**0.05**	0.08	0.06
Standard deviation	**0.05**	**0.05**	0.07	0.11	**0.05**	0.09	0.06
Max	0.34	**0.33**	0.45	1.01	0.40	0.61	0.35
Min	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Measurement	Sharpe ratio						
Algorithm	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{RoR}$	MOO_{Tr}	MOO_{Ror}	MOO_{Risk}	MOO_{TrRor}	MOO_{TrRisk}
	0.82	0.67	0.86	0.62	0.49	0.76	**0.89**

the corresponding fitness function.

The experiments, conducted on 220 datasets from 10 different international markets, showed that the proposed αMOO2 algorithms can significantly outperform the single-objective GP in terms of expected RoR without compromising on risk, yet also demonstrate the superiority of the proposed MOO3 algorithm. The MOO3 algorithm effectively generated a diverse set of Pareto-optimal solutions that provided optimal trade-offs among the three objectives. Statistical analysis further reinforced our findings. Kolmogorov-Smirnov tests indicated significant differences arising between MOO3 and SOO in several cases, while the non-parametric Friedman test with Hommel's post-hoc analysis showed that MOO3 algorithms focusing on individual objectives performed best, followed by those combining multiple objectives. Kolmogorov-Smirnov tests also indicated that proposed αMOO2 algorithms achieved significant improvement in terms of $\mathbb{E}[RoR]$ with the same level of risk as compared to the SOO algorithms. These results highlight the benefits of considering multiple objectives under a MOO framework. Moreover, the proposed MOO algorithms were benchmarked against traditional technical indicators (MACD, OBV, and MTM) and the buy-and-hold strategy. The

Table 6.29: Comparison between MOO2 and MOO3 algorithms on 10-year periods. Best values per row appear in boldface.

Measurement	Total return						
Algorithm	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{RoR}$	MOO_{Tr}	MOO_{Ror}	MOO_{Risk}	MOO_{TrRor}	MOO_{TrRisk}
Average	32.04%	32.50%	**61.33%**	48.29%	15.15%	51.90%	41.58%
Median	17.33%	19.02%	**41.68%**	32.81%	10.06%	37.88%	24.18%
Standard deviation	0.53	0.46	0.67	0.69	**0.24**	0.61	0.52
Max	352.30%	294.70%	341.71%	**544.98%**	152.06%	324.26%	258.04%
Min	-38.16%	**-31.20**%	-44.63%	-44.89%	-32.11%	-42.72%	-37.47%
Measurement	Expected Rate of Return						
Algorithm	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{RoR}$	MOO_{Tr}	MOO_{Ror}	MOO_{Risk}	MOO_{TrRor}	MOO_{TrRisk}
Average	1.67%	1.84%	3.28%	**3.38%**	0.99%	2.76%	1.63%
Median	1.66%	1.68%	2.52%	**2.62%**	0.86%	2.38%	1.44%
Standard deviation	**0.01**	0.02	0.05	0.04	0.02	0.02	**0.01**
Max	7.98%	7.21%	**44.36%**	25.92%	12.24%	13.07%	6.80%
Min	-**1.91**%	-2.18%	-4.03%	-4.28%	-1.98%	-2.98%	-1.23%
Measurement	Risk						
Algorithm	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{RoR}$	MOO_{Tr}	MOO_{Ror}	MOO_{Risk}	MOO_{TrRor}	MOO_{TrRisk}
Average	0.07	0.07	0.08	0.09	**0.05**	0.09	0.06
Median	0.06	0.06	0.06	0.07	**0.03**	0.07	0.05
Standard deviation	0.05	0.05	0.06	0.07	0.05	0.06	**0.04**
Max	0.33	0.29	0.43	0.49	0.42	0.41	**0.29**
Min	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Measurement	Sharpe ratio						
Algorithm	$\alpha MOO2^{Sig}_{RoR}$	$\alpha MOO2^{Cos}_{RoR}$	MOO_{Tr}	MOO_{Ror}	MOO_{Risk}	MOO_{TrRor}	MOO_{TrRisk}
	1.11	1.08	0.66	0.76	0.65	1.10	**1.12**

proposed MOO algorithms consistently outperformed these benchmarks, demonstrating their robustness and efficiency in optimising trading strategies. In the last section, we compare the MOO2 algorithms with the MOO3 algorithm in terms of TR, $\mathbb{E}[RoR]$, and risk. The results suggest that the MOO3 algorithms, specifically MOO_{TrRor}, exhibit better performance than MOO2 algorithms.

In summary, this chapter provides evidence that MOO using the combination of physical time and DC indicators offers significant improvements over SOO. The proposed algorithms not only produce profitable trading strategies, but are also able to consider the variety of investor preferences, making them a useful tool for algorithmic trading.

Table 6.30: Friedman with Bonferroni's post-hoc test between MOO3 algorithms and MOO2 algorithms across a 5-year period. Statistically significant differences at the 5% level are shown in boldface.

Table 6.31: Total return

Algorithm	Average rank	p_{Bonf}
MOO_{Tr}(c)	2.790909	-
MOO_{TrRor}	3.018182	1.04E-01
MOO_{Ror}	3.509091	**4.43E-03**
MOO_{TrRisk}	3.800000	**2.43E-04**
$\alpha MOO2^{Sig}_{RoR}$	4.390909	**4.35E-08**
$\alpha MOO2^{Cos}_{RoR}$	4.627273	**5.85E-10**
MOO_{Risk}	5.809091	**1.77E-22**

Table 6.32: Expected Rate of Return

Algorithm	Average rank	p_{Bonf}
MOO_{TrRor}(c)	2.972727	-
MOO_{Tr}	3.000000	2.57E-01
MOO_{Ror}	3.181818	1.60E-01
$\alpha MOO2^{Sig}_{RoR}$	4.218182	**1.00E-05**
MOO_{TrRisk}	4.354545	**2.01E-06**
$\alpha MOO2^{Cos}_{RoR}$	4.481818	**3.21E-07**
MOO_{Risk}	5.736364	**2.42E-19**

Table 6.33: Risk

Algorithm	Average rank	p_{Bonf}
MOO_{Risk}(c)	1.818182	-
MOO_{TrRisk}	2.945455	**1.91E-05**
$\alpha MOO2^{Sig}_{RoR}$	3.809091	**6.00E-12**
$\alpha MOO2^{Cos}_{RoR}$	3.854545	**2.52E-12**
MOO_{Tr}	4.636364	**5.50E-21**
MOO_{Ror}	5.318182	**3.37E-30**
MOO_{TrRor}	5.590909	**5.17E-34**

Table 6.34: Friedman with Bonferroni's post-hoc test between MOO3 algorithms and MOO2 algorithms over the 10-year period. Statistically significant differences at the 5% level are shown in boldface.

Table 6.35: Total return

Algorithm	Average rank	p_{Bonf}
MOO_{Tr}(c)	2.127273	-
MOO_{TrRor}	3.072727	**3.62E-04**
MOO_{Ror}	3.663636	**8.66E-08**
MOO_{TrRisk}	4.009091	**1.29E-10**
$\alpha MOO2^{Cos}_{RoR}$	4.554545	**5.48E-16**
$\alpha MOO2^{Sig}_{RoR}$	4.663636	**4.03E-17**
MOO_{Risk}	5.909091	**1.06E-33**

Table 6.36: Expected Rate of Return

Algorithm	Average rank	p_{Bonf}
MOO_{Ror}(c)	2.772727	-
MOO_{TrRor}	2.772727	2.86E-01
MOO_{Tr}	2.900000	2.06E-01
$\alpha MOO2^{Cos}_{RoR}$	4.300000	**1.45E-07**
$\alpha MOO2^{Sig}_{RoR}$	4.590909	**7.35E-10**
MOO_{TrRisk}	4.709091	**8.42E-11**
MOO_{Risk}	5.954545	**4.89E-25**

Table 6.37: Risk

Algorithm	Average rank	p_{Bonf}
MOO_{Risk}(c)	1.345455	-
MOO_{TrRisk}	2.990909	**7.85E-09**
$\alpha MOO2^{Cos}_{RoR}$	3.827273	**5.58E-17**
$\alpha MOO2^{Sig}_{RoR}$	4.081818	**8.71E-20**
MOO_{Tr}	4.481818	**6.88E-25**
MOO_{Ror}	5.572727	**3.24E-41**
MOO_{TrRor}	5.700000	**5.14E-43**

Chapter 7

Thesis Contributions

Throughout this thesis, we have focused on the application of ML algorithms, specifically GP and NSGA-II algorithms under the DC framework in the field of algorithmic trading. More specifically, we investigated whether the DC approach can be competitive with the traditional physical time approach. To achieve this goal, we adopted a variety of datasets, exploring novel trading strategies under the DC framework, a combination of PT and DC indicators, and even using a MOO algorithm. In this chapter, we present the main contributions from the preceding chapters.

7.1 Summary of application on physical time scale

7.1.1 Motivation

In Chapter 4, we conducted an in-depth comparison of a GP algorithm against nine different ML algorithms, highlighting the advantages of the GP algorithm when utilising technical indicators. According to the literature, ML techniques have proven to be valuable tools in financial forecasting, while GP algorithms have demonstrated success in combining diverse indicators for profitable trading strategies. However, previous experiments have typically used fewer than four benchmarks when evaluating GP performance. In addition, we also noticed that some factors that influence the performance of the approach were often neglected, including the number of markets, transaction costs, data periods, and different forecasting

days ahead.

7.1.2 Originality

This work conducted a comprehensive comparison of a GP algorithm against nine different ML algorithms, considering factors such as the variety of stock markets, transaction costs, variant data periods, and divergent forecasting horizons.

7.1.3 Findings

The results have demonstrated that the GP algorithm statistically outperformed most algorithms in terms of risk, while also yielding profitable outcomes. This is a significant finding, as previous studies have typically focused on fewer ML algorithms and/or fewer datasets. Further analysis revealed differences in performance across various international indices and market conditions.

7.2 Summary of application to event-based framework

7.2.1 Motivation

In the previous experiments, the GP-based algorithm successfully outperforms most benchmarks. However, it also was outperformed by the SVA algorithm, which motivates us to seek further improvements. One promising approach is DC, an alternative method that summarises stock movements as a series of events. However, most DC applications have focused on the Forex market to date, leaving a gap in research regarding its application to the stock market. Additionally, to the best of our knowledge, no research has utilised a large number of indicators in financial forecasting within the DC framework. Previous studies have indicated that GP algorithms are effective in combining diverse indicators to create profitable trading strategies. Therefore, we proposed using GP-based algorithms to explore whether DC could be competitive with the physical time method. Moreover, we investigated whether integrating both DC and physical time indicators could further enhance financial performance.

7.2.2 Originality

This work proposed two GP-based algorithms, GP-PT and GP-DC, which incorporated 28 physical time indicators and DC indicators, respectively. Moreover, in order to evaluate whether these two families of indicators are complementary, we also created another trading strategy containing 28 DC and 28 physical time indicators. This gave rise to the GP-DC-PT algorithm and the ensuing experiments were conducted across a variety of stock markets.

7.2.3 Findings

In conclusion, the proposed DC-based algorithms achieved profitable results that outperformed the traditional physical time strategy (GP-PT), while maintaining relatively low risk. Furthermore, the combined method (GP-DC-PT) successfully enhanced algorithmic trading performance, surpassing both another DC-based algorithm (GP-DC) as well as the non-GP benchmarks. Therefore, we argue that the DC approach can not only be a strong competitor to the traditional physical time approach but also enhance performance when used in conjunction with the physical time approach.

7.3 Summary of application of multi-objective optimisation

7.3.1 Motivation

In the previous chapter, the proposed GP-based algorithms demonstrated significant performance, exploring the profitable potential of the DC approach. However, we observed the solutions generated by GP-based algorithms showed passive trading behaviour. This was caused by the Sharpe ratio as an aggregate fitness function, which is calculated based on the expected RoR and risk. GP could easily achieve low risk by maintaining few trades, consequently achieving a high Sharpe ratio without necessarily optimising the expected RoR. To overcome the problem, we extended our work into the MOO field.

In Chapter 6, we investigated the MOO2 model, which optimises the expected RoR and risk, and the MOO3 model, which optimises the total return, expected RoR, and risk. These

MOO algorithms all utilised the well-known NSGA-II. Given the similarity between GP and GA, the NSGA-II approach we employed was based on the GP process. Additionally, we incorporated the α dominance strategy into MOO2 to address the challenges associated with the convergence ability of multi-objective GP.

7.3.2 Originality

This work addressed the gap in MOO applications within the DC framework. We explored the benefits of the MOO algorithm under the DC framework by benchmarking against SOO using the same indicators. In addition, considering the needs of the traders in real life, we defined a way to select a single 'best' solution from the Pareto front generated by the MOO approach, based on the performance of the corresponding fitness function in the training dataset.

7.3.3 Findings

The experiments, conducted using the same datasets and trading strategies as in the previous chapters, demonstrated that both the MOO2 and MOO3 approaches outperform the SOO approach. It is noteworthy that the results of the MOO algorithms are derived from the 'best' solution on the Pareto front which indicates profitability. This strongly suggests that MOO, specifically MOO3, within the DC framework significantly enhances performance when compared to SOO. The proposed algorithms not only yield superior trading strategies but also accommodate diverse investor preferences, making them potentially valuable tools for algorithmic trading.

7.4 Future work

Our work provides evidence that the DC, which summarises stock market movement from an alternative view, is able to enhance the traditional tools in SOO and MOO. Future research in this area could explore several promising directions to further enhance the effectiveness and robustness of algorithmic trading within the DC framework.

The first potential direction involves employing an approach that distinguishes the search

for DC and physical time (PT) indicators using strongly-typed GP. In the current study, these indicators were directly adopted to train the GP algorithm, resulting in GP trees that integrate both PT and DC indicators randomly. Strongly-typed GP, however, enforces type constraints on genetic operations, enabling the generation of trees where one branch contains PT indicators only and another contains DC indicators only. This approach allows for an in-depth investigation of the information derived from both PT and DC frameworks by generating a tree with one subtree created from PT indicators and another from DC indicators.

A second potential direction involves developing more sophisticated methods for selecting a single trading strategy from the Pareto front. In this thesis, the 'best' trading strategy was chosen from the Pareto front based on its performance on the training set to meet traders' needs. The metrics considered were either the Sharpe ratio or one of the following: total return, expected RoR, and risk. However, using the single 'best' solution loses the advantage of a MOO algorithm on the concept of the Pareto front. Future research could incorporate weighting schemes for several solutions within the Pareto front instead of using single 'best' solutions. This approach would allow for the development of a complex trading strategy comprising several solutions with different weights allocated by another algorithm. Such a strategy could provide richer information and spread risk, thereby enhancing our understanding of the application of MOO in real-life scenarios.

The third potential direction is the adoption of more complex genetic operators and trading strategies. This thesis employed four operators: AND, OR, $<$, and $>$. However, as a symbolic regression model, GP has the capability to generate more comprehensive solutions using additional operators and novel approaches. For example, it can incorporate scaling laws with DC indicators within the DC framework as part of its terminal set to improve financial forecasting performance. Scaling laws were explored and favoured within the domain of DC. There was much DC literature that discovered and used the profitable scaling laws in [125, 28, 126].

Additionally, the current trading strategy was designed to answer a simple question: whether the price will increase by a given percentage within a predefined time. When the GP returns 'True', we buy one amount of stock, otherwise, hold. During the whole trading

strategy, we either hold one amount of stock or nothing. This simplistic strategy was adopted because this thesis focuses on the comparison between algorithms. Future work could explore more complex trading strategies to explore the potential of GP. For example, we could add the amount of buying stock or limit the cost on the first day as an additional parameter to build a new trading strategy, thereby simulating real trading.

Chapter 8

Appendix A

8.1 Physical time indicators

In this appendix, we introduce the physical time indicators used in Chapter 4, which were not previously presented.

1) *Stochastic %K*

   Stochastic %K represents the current price of the security. The formula is presented in Equation A1.

$$Stochastic\ \%k = 100 - William's\ \%R \tag{A1}$$

   where the calculation of William's %R could be found in Equation 2.6.

2) *Stochastic %D* Stochastic %D is calculated as the moving average of the Stochastic %K.

3) *Momentum*

   Momentum (MTM) is a simple leading momentum indicator. It measures the rate at which a trend is accelerating or decelerating. The formula is presented in A2.

$$MTM = C_{i-1} - C_i \tag{A2}$$

   where C_{i-1} represents the closing price of the previous day and C is the current closing price.

4) *Relative Difference in percentage*

The relative difference in percentage (RDP) is the relative percentage difference arising between two points. In our experiments, we calculated it between the current point with a data point in n days ahead. The formula is presented in Equation A3.

$$RDP = \frac{C_{i-n} - C_i}{C_i} \tag{A3}$$

where C is the closing price.

5) *Rate of Change*

Rate of change (ROC) measures the speed at which the stock price changes over a specific period. The formula is presented in Equation A4.

$$ROC = \frac{P_i}{P_{i-n}} \tag{A4}$$

where P_i stands for the price on day *i* and P_{i-n} stands for the price at n day before day *i*.

6) *Disparity index* The disparity index shows the relationship between a most recent price with a particular moving average. The formula is presented in Equation A5.

$$Disparity = \frac{P_i}{MA_n} \tag{A5}$$

where P_i represents the price on day *i* and MA_n is the moving average over n days, which is presented in Equation 2.2.

7) *Percent Price Oscillator*

Percent price oscillator (PPO) is a momentum indicator that looks at the relationship between two exponential moving averages (EMA) of an asset. The formula is presented in Equation A6.

$$PPO = \frac{(short - term\ EMA) - (long - term\ EMA)}{long - term\ EMA} \tag{A6}$$

where short-term EMA and long-term EMA represent the EMA over a shorter period and a longer period, respectively. The formula for EMA is presented in Equation 2.9.

8) *Aroon Indicator*

The Aroon indicator determines the changes happening in the uptrend or downtrend. It consists of Aroon Up and Aroon Down. The formulas are presented in Equation A7 and Equation A8.

$$Aroon\ Up = \frac{(Number\ of\ periods) - (Number\ of\ periods\ since\ high)}{Number\ of\ periods} \tag{A7}$$

$$Aroon\ Down = \frac{(Number\ of\ periods) - (Number\ of\ periods\ since\ low)}{Number\ of\ periods} \tag{A8}$$

where the number of periods represents the user-specific periods, and the number of periods, since high\low is the period after the highest\lowest price occurred.

9) *Chande Momentum Oscillator*

The Chande momentum oscillator (CMO) is a technical indicator that uses momentum to identify relative strengths in a market. The formulae are presented in Equation A9 and Equation A10.

$$CMO = \frac{(S_p) - (S_n)}{(S_p) + (S_n)} \tag{A9}$$

$$P_i = \begin{cases} UP, & \text{if } P_i > P_{i-1} \\ DOWN, & \text{if } P_i < P_{i-1} \end{cases} \tag{A10}$$

where P_i is the price on day i. S_p and S_n represent the sum of the up and down prices over a specific period.

10) *Money flow index*

The money flow (MF) index identifies the overbought and oversold signal in an asset by using price and column data. The formulae are presented from Equation A11 to Equation A14.

$$MF = \frac{100}{1 + Money\ Flow\ Ratio} \tag{A11}$$

$$Money\ Flow\ Ratio = \frac{Sum\ Of\ Positive\ Money\ Flow}{Sum\ Of\ Negative\ Money\ Flow} \tag{A12}$$

$$Money\ Flow = Typical\ price \times Volume \tag{A13}$$

$$Typical\ price = \frac{(H + L + C)}{3} \tag{A14}$$

where H, L, and C represent the high, low, and closing prices. The positive\ negative money flow is defined when the current money flow is higher\lower than the previous money flow.

11) *Donchian Channel*

Donchian channel is a technical indicator used to determine the relative volatility of a market. The formula is presented in Equation A15.

$$DC = \frac{(HH + LL)}{2} \tag{A15}$$

where HH represents the highest high price over a specific period and LL represents the lowest low price over a specific period.

12) *Relative Volatility Index*

The relative volatility index (RVI) is a volatility indicator, similar to the relative strength

index (RSI), although it has a few key differences. The RVI measures the standard deviation of prices as they change over time. The formulae are presented from Equation A16 to Equation A20.

$$RVI = \frac{(UPavg)}{UPavg + DOWNavg} \tag{A16}$$

$$UPavg = \frac{(UPavg \times (n-1) + UP)}{n} \tag{A17}$$

$$DOWNavg = \frac{(DOWNavg \times (n-1) + DOWN)}{n} \tag{A18}$$

$$UP_i = \begin{cases} StdDev_{i-9}^{i}(P_i), & \text{if } P_i > P_{i-1} \\ 0, & \text{else} \end{cases} \tag{A19}$$

$$DOWN_i = \begin{cases} StdDev_{i-9}^{i}(P_i), & \text{if } P_i < P_{i-1} \\ 0, & \text{else} \end{cases} \tag{A20}$$

where P_i represents the price on day *i*.

13) *Accumulation/Distribution*

The accumulation/distribution (AD) line indicates the supply and demand of an asset using the closing price and volume. The formula is presented in Equation A21.

$$AD = \frac{(C-O)}{H-L} \times volume \tag{A21}$$

where C and O represent the closing and open price, respectively, and H and L are the high and low prices.

14) *De-trended Price Oscillator*

The de-trended price oscillator (DPO) is a technical indicator that measures the length of the price cycle. The formula is presented in Equation A22.

$$DPO = P_{i-\frac{n}{2}+1} - MA(n) \tag{A22}$$

where n represents the number of days used for the look-back period and MA is the moving average presented in Equation 2.2.

15) *Double Exponential Moving Average*

The double exponential moving average (DEMA) is an improvement on the EMA, allocating more weight to recent data. The formula is presented in Equation A23.

$$DEMA = 2 \times EMA(C, n) - EMA(EMA(C, n), n) \tag{A23}$$

where n represents the look-back period and C is the closing price. The formula of EMA is presented in Equation 2.9.

16) *Directional Movement Index*

The directional movement index (DMI) helps traders to determine the relative strength and direction of price trends. The formulas are presented from Equation A24 to Equation A29.

$$DMI = \frac{|+DI - -DI|}{|+DI + -DI|} \tag{A24}$$

$$+DI = \frac{\sum_{i=1}^{n} +DM - \frac{\sum_{i=1}^{n} +DM}{n} + +DM}{\sum_{i=1}^{n} TR - \frac{\sum_{i=1}^{n} TR}{n} + TR} \tag{A25}$$

$$-DI = \frac{\sum_{i=1}^{n} -DM - \frac{\sum_{i=1}^{n} -DM}{n} + -DM}{\sum_{i=1}^{n} TR - \frac{\sum_{i=1}^{n} TR}{n} + TR} \tag{A26}$$

$$MIN[+DM, -DM] = 0 \tag{A27}$$

$$+DM = \begin{cases} H_i - H_{i-1}, & \text{if } H_i > H_{i-1} \\ 0, & \text{else} \end{cases} \tag{A28}$$

$$-DM = \begin{cases} L_{i-1} - L_i, & \text{if } L_{i-1} > L_i \\ 0, & \text{else} \end{cases} \tag{A29}$$

where the formula for TR is presented in Equation 2.8, n represents the look-back period, and H and L are the high price and low price, respectively.

17) *Ease of Movement*

Ease of Movement (EOM) is a technical indicator that measures the price trend using momentum and volume information. The formulae are presented in Equations A30 and A31.

$$EOM = \frac{\frac{H_i - L_i}{2} - \frac{H_{i-1} - L_{i-1}}{2}}{Box\ Ratio} \tag{A30}$$

$$Box\ Ratio = \frac{\frac{volume}{10000}}{H - L} \tag{A31}$$

where H and L represent the high price and low price, respectively.

18) *Klinger Oscillator*

The Klinger oscillator (KO) is a technical indicator that measures the long-term trend of the money flow, thereby maintaining the sensitivity to short-term movement. The formulas are presented from Equation A32 to Equation A37.

$$KO = VFEMA_i(34) - VFEMA_i(55) \tag{A32}$$

$$VFEMA = VF \times \frac{2}{1+n} + EMA_{i-1} \times (1 - \frac{2}{1+n}) \tag{A33}$$

$$VF_i = volume_i \times \left| 2 \times (\frac{dm_i}{cm_i} - 1) \right| \times trend_i \times 100 \tag{A34}$$

$$trend_i = \begin{cases} 1, & \text{if } (H_i + L_i + C_i)) > (H_{i-1} + L_{i-1} + C_{i-1}) \\ -1, & \text{else} \end{cases} \tag{A35}$$

$$cm_i = \begin{cases} cm_{i-1} + dm_i, & \text{if } trend_i = trend_{i-1} \\ dm_{i-1} + dm_i, & \text{else} \end{cases} \tag{A36}$$

$$dm_i = H_i - L_i \tag{A37}$$

where dm represents the daily measurement and cm is the cumulative measurement. The vf represents the volume force. The H, L, and C stand for the high, low, and closing price.

19) *Market Facilitation Index*

The market facilitation index (MFI) calculates the price movement per volume to determine the effectiveness of price movement. The formula is presented in Equation A38.

$$MFI = \frac{H_i - L_i}{volume_i} \tag{A38}$$

where H and L represent the high and low prices.

20) *Mass Index*

Mass index (MI) is a technical indicator used to predict trend reversals. The formula is presented in Equation A39.

$$MI = \sum_{i=1}^{25} \frac{EMA((H-L), 9)}{EMA(EMA((H-L), 9), 9)} \tag{A39}$$

where EMA((H-L),9) represents an exponential moving average (EMA) which is calculated by (high price-low price) with a period of 9. The formula for EMA is presented in

Equation 2.9.

21) *Negative Volume Index*

The negative volume index (NVI) is a technical indication line that shows how the price movement is affected by down volume days. The formula is presented in Equation A40.

$$NVI_i = \begin{cases} NVI_{i-1} + \frac{C_i - C_{i-1}}{C_{i-1}}, & \text{if } volume_i < volume_{i-1} \\ NVI_{i-1}, & \text{else} \end{cases} \tag{A40}$$

where C represents the closing price.

22) *Percentage Volume Oscillator*

The percentage volume oscillator (PVO) is a technical indicator that provides traders the insight into the momentum of volume changes. The formula is presented in Equation A41.

$$PVO = 100 \times \frac{fastEMA(volume) - slowEMA(volume)}{fastEMA(volume)} \tag{A41}$$

where fastEMA and slowEMA represent an exponential moving average (EMA) with a longer period and a shorter period, respectively.

22) *Polarized Fractal Efficiency*

Polarized fractal efficiency (PFE) is used to measure the price efficiency of an investment. Traders can use PFE to check the price trend of the stock. The formulae are presented in Equation A42 and Equation A43.

$$PFE = EMA(K_i, n) \tag{A42}$$

$$K_i = 100 \times \frac{\sqrt{(P_i - P_{i-m})^2 + m^2}}{\sum_{j=0}^{m-2} \sqrt{(P_{i-j} - P_{i-j-1})^2 + 1}} \tag{A43}$$

where m represents the period of the PFE and n is the smoothing period. P_i stands for the closing price on day *i*.

23) *Random Walk Index*

The random walk index (RWI) compares the price movement to random movement to determine whether the stock is in a significant trend. There are two lines of RWIhigh and RWIlow, which are presented in Equation A44 and Equation A45.

$$RWIhigh = \frac{H_{i-n+1} - L_i}{ATR_i \times \sqrt{n}} \tag{A44}$$

$$RWIlow = \frac{H_i - L_{i-n+1}}{ATR_i \times \sqrt{n}} \tag{A45}$$

where ATR represents the average true range in Equation 2.7. H and L are the high price and low price, and *n* represents the look-back period of the RWI.

24) *Relative Momentum Index*

The relative momentum index (RMI) measures the speed and magnitude of the stock price movement. The formulae are presented from Equation A46 to Equation A50.

$$RMI = 100 \times \frac{upavg}{upavg + dnavg} \tag{A46}$$

$$UPavg = \frac{(UPavg \times (n-1) + UP)}{n} \tag{A47}$$

$$DOWNavg = \frac{(DOWNavg \times (n-1) + DOWN)}{n} \tag{A48}$$

$$UP = \begin{cases} C_i - C_{i-m}, & \text{if } C_i > C_{i-m} \\ 0, & \text{else} \end{cases} \tag{A49}$$

$$DOWN = \begin{cases} C_{i-m} - C_i, & \text{if } C_i < C_{i-m} \\ 0, & \text{else} \end{cases} \tag{A50}$$

where C is the closing price, and M and n represent the periods of the RWI.

25) *Stochastic Momentum Index*

The stochastic momentum index (SMI) provides information for identifying overbought or oversold conditions for traders. The formulae are presented in Equations A51 to A53.

$$SMI = 100 \times \frac{EMA(EMA(cm, n), n)}{\frac{EMA(EMA(hl,n),n)}{2}} \tag{A51}$$

$$cm_i = C_i - \frac{HH_i + LL_i}{2} \tag{A52}$$

$$hl_i = HH_i - LL_i \tag{A53}$$

where HH and LL stand for the highest high price and lowest low price over a specific period, and *n* represents the period of the SMI.

26) *Triple Exponential Average*

The triple exponential average (TRIX) is a momentum indicator that measures the percentage change in a moving average that has been smoothed exponentially three times. The formulae are presented in Equations A54 and A55.

$$TRIX = 100 \times \frac{M_i - M_{i-1}}{M_i} \tag{A54}$$

$$M = EMA(EMA(EMA(P, n), n), n) \tag{A55}$$

where EMA is the exponential moving average delineated in Equation 2.9.

27) *Vertical Horizontal Filter*

The vertical horizontal filter (VHF) is a technical analysis tool that helps traders and investors identify the trend of price movements. The formula is presented in Equation A56.

$$VHF = 100 \times \frac{HH - LL}{\sum_{i=1}^{n} |C_i - C_{i-1}|} \tag{A56}$$

where HH and LL are the highest high price and lowest low price over a specific period, and *n* is the period of the VHF.

28) *Volume-Adjusted Moving Average*

The volume-adjusted moving average (VAMA) takes price and volume as being of equivalent weight when computing the moving average. The formula is presented in Equation A57.

$$VAMA = \frac{\sum_{i=0}^{(} n)(C_i \times volume_i)}{\sum_{i=0}^{(} n)(volume_i)} \tag{A57}$$

where C represent the closing price and *n* is the period of VAMA.

29) *Moving Average Convergence/Divergence (MACD)*

The moving average convergence/divergence (MACD) is a popular momentum indicator that identifies potential buy and sell signals in the stock market. The formula is presented in Equation A58.

$$MACD = EMA(C, n) - EMA(C, m) \tag{A58}$$

where C represents the closing price, and *n* and *m* stand for the periods of the EMA, respectively, with *m* being a longer period than *n*.

Bibliography

[1] Xinpeng Long, Michael Kampouridis, and Delaram Jarchi. An in-depth investigation of genetic programming and nine other machine learning algorithms in a financial forecasting problem. In *2022 IEEE Congress on Evolutionary Computation (CEC)*, pages 01–08. IEEE, 2022.

[2] Xinpeng Long, Michael Kampouridis, and Panagiotis Kanellopoulos. Genetic programming for combining directional changes indicators in international stock markets. In *Parallel Problem Solving from Nature–PPSN XVII: 17th International Conference, PPSN 2022, Dortmund, Germany, September 10–14, 2022, Proceedings, Part II*, pages 33–47. Springer, 2022.

[3] Xinpeng Long, Michael Kampouridis, and Panagiotis Kanellopoulos. Multi-objective optimisation and genetic programming for trading by combining directional changes and technical indicators. In *2023 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2023.

[4] Xinpeng Long and Michael. Kampouridis. α-dominance two-objective optimization genetic programming for algorithmic trading under a directional changes environment. In *2024 IEEE Conference on Computational Intelligence for Financial Engineering & Risk (CIFEr)*, pages 1–8. IEEE, 2024.

[5] Monira Essa Aloud. Time series analysis indicators under directional changes: The case of saudi stock market. *International Journal of Economics and Financial Issues*, 6(1):55–64, 2016.

[6] Edward PK Tsang, Ran Tao, and Shuai Ma. Profiling financial market dynamics under directional changes. *Working Paper WP077-15 Centre for Computational Finance and Economic Agents (CCFEA), University of Essex, Tech. Rep.*, 2015.

[7] Amit Kelotra and Prateek Pandey. Stock market prediction using optimized deep-convlstm model. *Big Data*, 8(1):5–24, 2020.

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

[9] Edward Tsang. Directional changes, definitions. *Working Paper WP050-10 Centre for Computational Finance and Economic Agents (CCFEA), University of Essex Revised 1, Tech. Rep.*, 2010.

[10] Mario Graff, Hugo Jair Escalante, Fernando Ornelas-Tellez, and Eric S Tellez. Time series forecasting with genetic programming. *Natural Computing*, 16(1):165–174, 2017.

[11] Alexandre Pimenta, Ciniro AL Nametala, Frederico G Guimarães, and Eduardo G Carrano. An automated investing method for stock market based on multiobjective genetic programming. *Computational Economics*, 52(1):125–144, 2018.

[12] Michael Kampouridis, Shu-Heng Chen, and Edward Tsang. Microstructure dynamics and agent-based financial markets: Can dinosaurs return? *Advances in Complex Systems*, 15(supp02):1250060, 2012.

[13] Hitoshi Iba and Takashi Sasaki. Using genetic programming to predict financial data. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volume 1, pages 244–251. IEEE, 1999.

[14] Dominique M Guillaume, Michel M Dacorogna, Rakhal R Davé, Ulrich A Müller, Richard B Olsen, and Olivier V Pictet. From the bird's eye to the microscope: A sur-

vey of new stylized facts of the intra-daily foreign exchange markets. *Finance and stochastics*, 1(2):95–129, 1997.

[15] Kokolo Ikeda, Hajime Kita, and Shigenobu Kobayashi. Failure of pareto-based moeas: Does non-dominated really mean near to optimal? In *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)*, volume 2, pages 957–962. IEEE, 2001.

[16] JG Agrawal, V Chourasia, and A Mittra. State-of-the-art in stock prediction techniques. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 2(4):1360–1366, 2013.

[17] Raymond Hon-fu Chan, Alan Wing-keung Wong, and Spike Tsz-ho Lee. *Technical analysis and financial asset forecasting: From simple tools to advanced techniques*. World Scientific Publishing Company, 2014.

[18] Christopher J Neely and Paul A Weller. Technical analysis in the foreign exchange market. *Federal Reserve Bank of St. Louis Working Paper No*, 2011.

[19] Eugene F Fama and Marshall E Blume. Filter rules and stock-market trading. *The Journal of Business*, 39(1):226–241, 1966.

[20] Panagiotis Rousis and Spyros Papathanasiou. Is technical analysis profitable on athens stock exchange? *Mega Journal of Business Research*, 2018, 2018.

[21] Cheol-Ho Park and Scott H Irwin. What do we know about the profitability of technical analysis? *Journal of Economic surveys*, 21(4):786–826, 2007.

[22] Preeti Baser, Jatinderkumar R Saini, and Narayan Baser. Gold commodity price prediction using tree-based prediction models. *International Journal of Intelligent Systems and Applications in Engineering*, 11(1s):90–96, 2023.

[23] AI Diler. Predicting direction of ise national-100 index with back propagation trained neural network. *Journal of Istanbul Stock Exchange*, 7(25-26):65–81, 2003.

[24] J Welles Wilder. *New concepts in technical trading systems*. Greensboro, NC, 1978.

[25] Benoit Mandelbrot and Howard M Taylor. On the distribution of stock price differences. *Operations research*, 15(6):1057–1062, 1967.

[26] S Raftopoulos. The zigzag trend indicator. *Technical analysis of stocks and commodities-magazine edition-*, 21(11):26–33, 2003.

[27] Bao Yang, Mouming Zhao, John Shi, Ning Yang, and Yueming Jiang. Effect of ultrasonic treatment on the recovery and dpph radical scavenging activity of polysaccharides from longan fruit pericarp. *Food chemistry*, 106(2):685–690, 2008.

[28] James B Glattfelder, Alexandre Dupuis, and Richard B Olsen. Patterns in high-frequency fx data: discovery of 12 empirical scaling laws. *Quantitative Finance*, 11(4):599–614, 2011.

[29] John R Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.

[30] Markus Brameier, Wolfgang Banzhaf, and Wolfgang Banzhaf. *Linear genetic programming*, volume 1. Springer, 2007.

[31] Julian Francis Miller and Simon L Harding. Cartesian genetic programming. In *Proceedings of the 11th annual conference companion on genetic and evolutionary computation conference: late breaking papers*, pages 3489–3512, 2009.

[32] Astro Teller and Manuela Veloso. Program evolution for data mining. *International Journal of Expert Systems Research and Applications*, 8(3):213–236, 1995.

[33] KA De Jong and W Spears. On the virtues of parameterized uniform crossover. In *Proceedings of the 4th Intern. Conf. on Genetic Algorithms, Morgan Kaufmann*, 1991.

[34] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, New York, 1989.

[35] William B Langdon et al. Evolving data structures with genetic programming. In *ICGA*, pages 295–302, 1995.

[36] Gregory S Hornby. Alps: the age-layered population structure for reducing the problem of premature convergence. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 815–822, 2006.

[37] William F Sharpe. The sharpe ratio. *Journal of portfolio management*, 21(1):49–58, 1994.

[38] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[39] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[40] Jialin Liu, Chih-Min Min Lin, and Fei Chao. Gradient boost with convolution neural network for stock forecast. In *Advances in Computational Intelligence Systems: Contributions Presented at the 19th UK Workshop on Computational Intelligence, September 4-6, 2019, Portsmouth, UK 19*, pages 155–165. Springer, 2020.

[41] Rebwar M Nabi, Saeed Soran Ab M, and Habibollah Harron. A novel approach for stock price prediction using gradient boosting machine with feature engineering (gbm-wfe). *Kurdistan Journal of Applied Research*, 5(1):28–48, 2020.

[42] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[43] Shangkun Deng, Xiaoru Huang, Yingke Zhu, Zhihao Su, Zhe Fu, and Tatsuro Shimada. Stock index direction forecasting using an explainable extreme gradient boosting and investor sentiments. *The North American Journal of Economics and Finance*, 64:101848, 2023.

[44] SFN Islam, A Sholahuddin, and AS Abdullah. Extreme gradient boosting (xgboost) method in making forecasting application and analysis of usd exchange rates against rupiah. In *Journal of Physics: Conference Series*, volume 1722, page 012016. IOP Publishing, 2021.

[45] Yongchang Lao, Fangzhong Qi, Jiakai Zhou, and Xiaobao Fang. A prediction method based on extreme gradient boosting tree model and its application. In *Journal of Physics: Conference Series*, volume 1995, page 012017. IOP Publishing, 2021.

[46] Jesse Davis, Laurens Devos, Sofie Reyners, and Wim Schoutens. Gradient boosting for quantitative finance. *Journal of Computational Finance*, 24(4), 2020.

[47] Sankeerth Rao Karingula, Nandini Ramanan, Rasool Tahmasbi, Mehrnaz Amjadi, Deokwoo Jung, Ricky Si, Charanraj Thimmisetty, Luisa F Polania, Marjorie Sayer, Jake Taylor, et al. Boosted embeddings for time-series forecasting. In *International Conference on Machine Learning, Optimization, and Data Science*, pages 1–14. Springer, 2021.

[48] Agustinus Bimo Gumelar, Haryati Setyorini, Derry Pramono Adi, Sengguruh Nilowardono, Agung Widodo, Achmad Teguh Wibowo, MY Teguh Sulistyono, Evy Christine, et al. Boosting the accuracy of stock market prediction using xgboost and long short-term memory. In *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, pages 609–613. IEEE, 2020.

[49] Pham Hoang Vuong, Trinh Tan Dat, Tieu Khoi Mai, Pham Hoang Uyen, et al. Stock-price forecasting based on xgboost and lstm. *Computer Systems Science & Engineering*, 40(1), 2022.

[50] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

[51] Osman Hegazy, Omar S Soliman, and Mustafa Abdul Salam. A machine learning model for stock market prediction. *arXiv preprint arXiv:1402.7351*, 2014.

[52] Rui Ren, Desheng Dash Wu, and Tianxiang Liu. Forecasting stock market movement direction using sentiment analysis and support vector machine. *IEEE Systems Journal*, 13(1):760–770, 2018.

[53] Huimin Tang, Peiwu Dong, and Yong Shi. A new approach of integrating piecewise linear representation and weighted support vector machine for forecasting stock turning points. *Applied soft computing*, 78:685–696, 2019.

[54] Nor Azizah Hitam, Amelia Ritahani Ismail, and Faisal Saeed. An optimized support vector machine (svm) based on particle swarm optimization (pso) for cryptocurrency forecasting. *Procedia computer science*, 163:427–433, 2019.

[55] Deepak Gupta, Mahardhika Pratama, Zhenyuan Ma, Jun Li, and Mukesh Prasad. Financial time series forecasting using twin support vector regression. *PloS one*, 14(3):e0211402, 2019.

[56] Anass Nahil and Abdelouahid Lyhyaoui. Short-term stock price forecasting using kernel principal component analysis and support vector machines: the case of casablanca stock exchange. *Procedia Computer Science*, 127:161–169, 2018.

[57] Aytaç Altan and Seçkin Karasu. The effect of kernel values in support vector machine to forecasting performance of financial time series. *The Journal of Cognitive Systems*, 4(1):17–21, 2019.

[58] Rongjun Yang, Lin Yu, Yuanjun Zhao, Hongxin Yu, Guiping Xu, Yiting Wu, and Zhengkai Liu. Big data analytics for financial market volatility forecast based on support vector machine. *International Journal of Information Management*, 50:452–462, 2020.

[59] Akshit Kurani, Pavan Doshi, Aarya Vakharia, and Manan Shah. A comprehensive comparative study of artificial neural network (ann) and support vector machines (svm) on stock forecasting. *Annals of Data Science*, 10(1):183–208, 2023.

[60] Robert A Dunne. *A statistical approach to neural networks for pattern recognition*. John Wiley & Sons, 2007.

[61] Qian Chen, Wenyu Zhang, and Yu Lou. Forecasting stock prices using a hybrid deep learning model integrating attention mechanism, multi-layer perceptron, and bidirectional long-short term memory neural network. *IEEE Access*, 8:117365–117376, 2020.

[62] Aditya Naik, Vijay Gaikwad, Rajesh Jalnekar, and Milind Rane. Daily stock price direction prediction using random multi-layer perceptron. In *2021 International Conference on Artificial Intelligence and Machine Vision (AIMV)*, pages 1–9. IEEE, 2021.

[63] Carlos Mate and Lucía Jimeńez. Forecasting exchange rates with the imlp: New empirical insight on one multi-layer perceptron for interval time series (its). *Engineering Applications of Artificial Intelligence*, 104:104358, 2021.

[64] Antonio Muñoz San Roque, Carlos Maté, Javier Arroyo, and Ángel Sarabia. imlp: Applying multi-layer perceptrons to interval-valued data. *Neural Processing Letters*, 25:157–169, 2007.

[65] Meenu Sreedharan, Ahmed M Khedr, and Magdi El Bannany. A multi-layer perceptron approach to financial distress prediction with genetic algorithm. *Automatic Control and Computer Sciences*, 54:475–482, 2020.

[66] Reza Ramezanian, Arsalan Peymanfar, and Seyed Babak Ebrahimi. An integrated framework of genetic network programming and multi-layer perceptron neural network for prediction of daily stock return: An application in tehran stock exchange market. *Applied soft computing*, 82:105551, 2019.

[67] Kang Zhang, Guoqiang Zhong, Junyu Dong, Shengke Wang, and Yong Wang. Stock market prediction based on generative adversarial network. *Procedia computer science*, 147:400–406, 2019.

[68] Muhammad Athanabil Andi Fawazdhia, Indra Cahya Ramdani, Shinta Aulia Septiani, and Zani Anjani Rafsanjani Hsm. Comparison between multiple linear regression and

multi-layer perceptron neural network for predicting stocks from mining sector in lq45. In *AIP Conference Proceedings*, volume 2733. AIP Publishing, 2023.

[69] Abdu Masanawa Sagir and Saratha Sathasivan. The use of artificial neural network and multiple linear regressions for stock market forecasting. *MATEMATIKA: Malaysian Journal of Industrial and Applied Mathematics*, pages 1–10, 2017.

[70] Alireza Namdari and Zhaojun Steven Li. Integrating fundamental and technical analysis of stock market through multi-layer perceptron. In *2018 IEEE technology and engineering management conference (TEMSCON)*, pages 1–6. IEEE, 2018.

[71] Romi Fadillah Rahmat, Alvin Rizki, Adil Fahad Alharthi, and Rahmat Budiarto. Big data forecasting using evolving multi-layer perceptron. In *2016 4th Saudi International Conference on Information Technology (Big Data Analysis)(KACSTIT)*, pages 1–5. IEEE, 2016.

[72] Evelyn Fix. *Discriminatory analysis: nonparametric discrimination, consistency properties*, volume 1. USAF school of Aviation Medicine, 1985.

[73] Li Tang, Heping Pan, and Yiyong Yao. K-nearest neighbor regression with principal component analysis for financial time series prediction. In *Proceedings of the 2018 International Conference on Computing and Artificial Intelligence*, pages 127–131, 2018.

[74] Haitham Fawzy, El Houssainy A Rady, and Amal Mohamed Abdel Fattah. Comparison between support vector machines and k-nearest neighbor for time series forecasting. *J. Math. Comput. Sci.*, 10(6):2342–2359, 2020.

[75] Rudra Kalyan Nayak, SYH Pavitra, Ramamani Tripathy, and K Prathyusha. Forecasting foreign currency exchange price using long short-term memory with k-nearest neighbour method. *International Journal of Engineering and Advanced Technology (IJEAT)*, 9(2), 2019.

[76] Guancen Lin, Aijing Lin, and Jianing Cao. Multidimensional knn algorithm based on eemd and complexity measures in financial time series forecasting. *Expert Systems with Applications*, 168:114443, 2021.

[77] Sioiok Wong. Stock price prediction model based on the short-term trending of knn method. In *2020 7th International Conference on Information Science and Control Engineering (ICISCE)*, pages 1355–1360. IEEE, 2020.

[78] Julius Tanuwijaya and Seng Hansun. Lq45 stock index prediction using k-nearest neighbors regression. *International Journal of Recent Technology and Engineering*, 8(3):2388–2391, 2019.

[79] Qian Yunneng. A new stock price prediction model based on improved knn. In *2020 7th International Conference on Information Science and Control Engineering (ICISCE)*, pages 77–80. IEEE, 2020.

[80] Klender Cortez, Martha del Pilar Rodríguez-García, and Samuel Mongrut. Exchange market liquidity prediction with the k-nearest neighbor approach: Crypto vs. fiat currencies. *Mathematics*, 9(1):56, 2020.

[81] Shengtao Gao. Trend-based k-nearest neighbor algorithm in stock price prediction. In *3rd International Conference on Digital Economy and Computer Application (DECA 2023)*, pages 746–756. Atlantis Press, 2023.

[82] Shivam Agarwal. Data mining: Data mining concepts and techniques. In *2013 international conference on machine intelligence and research advancement*, pages 203–207. IEEE, 2013.

[83] EHA Rady, Haitham Fawzy, and Amal Mohamed Abdel Fattah. Time series forecasting using tree based methods. *J. Stat. Appl. Probab*, 10(1):229–244, 2021.

[84] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The m4 competition: Results, findings, conclusion and way forward. *International Journal of forecasting*, 34(4):802–808, 2018.

[85] Pujana Paliyawan. Stock market direction prediction using data mining classification. *future*, 5:6, 2006.

[86] G Vadivu Navin. Big data analytics for gold price forecasting based on decision tree algorithm and support vector regression (svr). *International Journal of Science and Research (IJSR)*, 4(3):2026–2030, 2015.

[87] Dipashree Patil, Shivani Patil, Shreya Patil, and Sandhya Arora. Financial forecasting of stock market using sentiment analysis and data analytics. In *Intelligent Sustainable Systems: Selected Papers of WorldS4 2021, Volume 2*, pages 423–430. Springer, 2022.

[88] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[89] Manoj Thakur and Deepak Kumar. A hybrid financial trading support system using multi-category classifiers and random forest. *Applied Soft Computing*, 67:337–349, 2018.

[90] Zheng Tan, Ziqin Yan, and Guangwei Zhu. Stock selection with random forest: An exploitation of excess return in the chinese stock market. *Heliyon*, 5(8), 2019.

[91] Christoph Lohrmann and Pasi Luukka. Classification of intraday s&p500 returns with a random forest. *International Journal of Forecasting*, 35(1):390–407, 2019.

[92] Suryoday Basak, Saibal Kar, Snehanshu Saha, Luckyson Khaidem, and Sudeepa Roy Dey. Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics and Finance*, 47:552–567, 2019.

[93] Despoina Makariou, Pauline Barrieu, and Yining Chen. A random forest based approach for predicting spreads in the primary catastrophe bond market. *Insurance: Mathematics and Economics*, 101:140–162, 2021.

[94] Sanjiban Sekhar Roy, Rohan Chopra, Kun Chang Lee, Concetto Spampinato, and Behnam Mohammadi-ivatlood. Random forest, gradient boosted machines and deep neural network for stock price forecasting: a comparative analysis on south korean

companies. *International Journal of Ad Hoc and Ubiquitous Computing*, 33(1):62–71, 2020.

[95] Subba Rao Polamuri, Kudipudi Srinivas, and A Krishna Mohan. Stock market prices prediction using random forest and extra tree regression. *Int. J. Recent Technol. Eng*, 8(1):1224–1228, 2019.

[96] Hyun Jun Park, Youngjun Kim, and Ha Young Kim. Stock market forecasting using a multi-task approach integrating long short-term memory and the random forest framework. *Applied Soft Computing*, 114:108106, 2022.

[97] Pushpendu Ghosh, Ariel Neufeld, and Jajati Keshari Sahoo. Forecasting directional movements of stock prices for intraday trading using lstm and random forests. *Finance Research Letters*, 46:102280, 2022.

[98] Yilin Ma, Ruizhu Han, and Xiaoling Fu. Stock prediction based on random forest and lstm neural network. In *2019 19th International Conference on Control, Automation and Systems (ICCAS)*, pages 126–130. IEEE, 2019.

[99] Kofi O Nti, Adebayo Adekoya, and Benjamin Weyori. Random forest based feature selection of macroeconomic variables for stock market prediction. *American Journal of Applied Sciences*, 16(7):200–212, 2019.

[100] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63:3–42, 2006.

[101] Yakub Kayode Saheed, Raji Mustafa Ayobami, and Terdoo Orje-Ishegh. A comparative study of regression analysis for modelling and prediction of bitcoin price. In *Blockchain Applications in the Smart Era*, pages 187–209. Springer, 2022.

[102] Ujjwal Pasupulety, Aiman Abdullah Anees, Subham Anmol, and Biju R Mohan. Predicting stock prices using ensemble learning and sentiment analysis. In *2019 IEEE second international conference on artificial intelligence and knowledge engineering (AIKE)*, pages 215–222. IEEE, 2019.

[103] Ernest Kwame Ampomah, Zhiguang Qin, and Gabriel Nyame. Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement. *Information*, 11(6):332, 2020.

[104] Christopher Neely, Paul Weller, and Rob Dittmar. Is technical analysis in the foreign exchange market profitable? a genetic programming approach. *Journal of financial and Quantitative Analysis*, 32(4):405–426, 1997.

[105] Marcos Alvarez Diaz. Speculative strategies in the foreign exchange market based on genetic programming predictions. *Applied financial economics*, 20(6):465–476, 2010.

[106] Adesola Adegboye, Michael Kampouridis, and Colin G Johnson. Regression genetic programming for estimating trend end in foreign exchange market. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 2017.

[107] Georgios A Vasilakis, Konstantinos A Theofilatos, Efstratios F Georgopoulos, Andreas Karathanasopoulos, and Spiros D Likothanassis. A genetic programming approach for eur/usd exchange rate forecasting and trading. *Computational economics*, 42:415–431, 2013.

[108] Lee A Becker and Mukund Seshadri. Gp-evolved technical trading rules can outperform buy and hold. In *3rd international workshop on computational intelligence in economics and finance*. Citeseer, 2003.

[109] Jun Wang. Trading and hedging in s&p 500 spot and futures markets using genetic programming. *Journal of Futures Markets: Futures, Options, and Other Derivative Products*, 20(10):911–942, 2000.

[110] Shusheng Ding, Tianxiang Cui, Xihan Xiong, and Ruibin Bai. Forecasting stock market return with nonlinearity: a genetic programming approach. *Journal of Ambient Intelligence and Humanized Computing*, 11:4927–4939, 2020.

[111] Michael Kampouridis and Fernando EB Otero. Heuristic procedures for improving the predictability of a genetic programming financial forecasting algorithm. *Soft Computing*, 21:295–310, 2017.

[112] Viktor Manahov and Hanxiong Zhang. Forecasting financial markets using high-frequency trading data: Examination with strongly typed genetic programming. *International Journal of Electronic Commerce*, 23(1):12–32, 2019.

[113] Viktor Manahov, Robert Hudson, and Hafiz Hoque. Return predictability and the 'wisdom of crowds': Genetic programming trading algorithms, the marginal trader hypothesis and the hayek hypothesis. *Journal of International Financial Markets, Institutions and Money*, 37:85–98, 2015.

[114] Michael Kampouridis and Fernando EB Otero. Evolving trading strategies using directional changes. *Expert Systems with Applications*, 73:145–160, 2017.

[115] Jin Li and Edward PK Tsang. Improving technical analysis predictions: An application of genetic programming. In *FLAIRS*, pages 108–112, 1999.

[116] Monira Aloud. A real-time adaptive bitcoin trading system using genetic programming. *Journal of Theoretical and Applied Information Technology*, 99(16):4153–4162, 2021.

[117] Eva Christodoulaki, Michael Kampouridis, and Panagiotis Kanellopoulos. Technical and sentiment analysis in financial forecasting with genetic programming. In *Proceedings of the IEEE Computational Intelligence for Financial Engineering and Economics (CIFEr)*. IEEE, 2022.

[118] Monira Aloud, Edward Tsang, Richard Olsen, and Alexandre Dupuis. A directional-change event approach for studying financial time series. *Economics*, 6(1):20120036, 2012.

[119] Amer Bakhach, Edward Tsang, Wing Lon Ng, and VL Raju Chinthalapati. Backlash agent: A trading strategy based on directional change. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–9. IEEE, 2016.

[120] Shaimaa Masry Hussein. *Event-based microscopic analysis of the fx market*. PhD thesis, University of Essex, 2013.

[121] Michael Kampouridis, Adesola Adegboye, and Colin Johnson. Evolving directional changes trading strategies with a new event-based indicator. In *Asia-Pacific Conference on Simulated Evolution and Learning*, pages 727–738. Springer, 2017.

[122] Monira Aloud, Edward Tsang, and Richard Olsen. Modeling the fx market traders' behavior: An agent-based approach. In *Banking, Finance, and Accounting: Concepts, Methodologies, Tools, and Applications*, pages 350–384. IGI Global, 2015.

[123] M Aloud. Directional-change event trading strategy: Profit-maximizing learning strategy. In *COGNITIVE 2015, The Seventh International Conference on Advanced Cognitive Technologies and Applications*, pages 123–129, 2015.

[124] Monira Essa Aloud. Profitability of directional change based trading strategies: The case of saudi stock market. *International Journal of Economics and Financial Issues*, 6(1):87–95, 2016.

[125] James B Glattfelder, A Dupuis, and R Olsen. An extensive set of scaling laws and the fx coastline. *Unpublished manuscript*, 2008.

[126] Monira Essa Aloud and Maria Fasli. Exploring trading strategies and their effects in the foreign exchange market. *Computational Intelligence*, 33(2), 2016.

[127] Han Ao and Edward Tsang. Trading algorithms built with directional changes. In *2019 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, pages 1–7. IEEE, 2019.

[128] Amer Bakhach, EP Tsang, and Wing Lon Ng. Forecasting directional changes in financial markets. *Working Paper WP075–15 Centre for Computational Finance and Economic Agents (CCFEA), University of Essex, Tech. Rep.*, 2015.

[129] Amer Bakhach, Edward PK Tsang, and Hamid Jalalian. Forecasting directional changes in the fx markets. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 2016.

[130] Nora Alkhamees and Maria Fasli. A directional change based trading strategy with dynamic thresholds. In *2017 IEEE international conference on data science and advanced analytics (DSAA)*, pages 283–292. IEEE, 2017.

[131] Nora Alkhamees and Maria Fasli. Event detection from time-series streams using directional change and dynamic thresholds. In *2017 IEEE international conference on big data (Big Data)*, pages 1882–1891. IEEE, 2017.

[132] Adesola Adegboye and Michael Kampouridis. Machine learning classification and regression models for predicting directional changes trend reversal in fx markets. *Expert Systems with Applications*, 173:114645, 2021.

[133] Adesola Adegboye, Michael Kampouridis, and Fernando Otero. Algorithmic trading with directional changes. *Artificial Intelligence Review*, 56(6):5619–5644, 2023.

[134] Edward PK Tsang, Ran Tao, Antoaneta Serguieva, and Shuai Ma. Profiling high-frequency equity price movements in directional changes. *Quantitative finance*, 17(2):217–225, 2017.

[135] Ahoora Rostamian and John G O'Hara. Event prediction within directional change framework using a cnn-lstm model. *Neural Computing and Applications*, 34(20):17193–17205, 2022.

[136] Ozgur Salman, Michael Kampouridis, and Delaram Jarchi. Trading strategies optimization by genetic algorithm under the directional changes paradigm. In *2022 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2022.

[137] Antonio C Briza and Prospero C Naval Jr. Design of stock trading system for historical market data using multiobjective particle swarm optimization of technical indicators.

In *Proceedings of the 10th annual conference companion on Genetic and evolutionary computation*, pages 1871–1878, 2008.

[138] Matthew Butler and Ali Daniyal. Multi-objective optimization with an evolutionary artificial neural network for financial forecasting. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 1451–1458, 2009.

[139] Dome Lohpetch and David Corne. Multiobjective algorithms for financial trading: Multiobjective out-trades single-objective. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 192–199. IEEE, 2011.

[140] Ricardo de Almeida, Gilberto Reynoso-Meza, and Maria Teresinha Arns Steiner. Multiobjective optimization approach to stock market technical indicators. In *2016 IEEE congress on evolutionary computation (CEC)*, pages 3670–3677. IEEE, 2016.

[141] Youngmin Kim and David Enke. Developing a rule change trading system for the futures market using rough set analysis. *Expert Systems with Applications*, 59:165–173, 2016.

[142] Frederick Ditliac Atiah and Mardé Helbig. Effects of decision models on dynamic multiobjective optimization algorithms for financial markets. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 762–770. IEEE, 2019.

[143] Seçkin Karasu, Aytaç Altan, Stelios Bekiros, and Wasim Ahmad. A new forecasting model with wrapper-based feature selection approach using multi-objective optimization technique for chaotic crude oil time series. *Energy*, 212:118750, 2020.

[144] Oscar Claveria, Enric Monte, and Salvador Torra. Evolutionary computation for macroeconomic forecasting. *Computational Economics*, 53(2):833–849, 2019.

[145] Sana Ben Hamida, Wafa Abdelmalek, and Fathi Abid. Applying dynamic training-subset selection methods using genetic programming for forecasting implied volatility. *arXiv preprint arXiv:2007.07207*, 2020.

[146] Erdinc Akyildirim, Aurelio F Bariviera, Duc Khuong Nguyen, and Ahmet Sensoy. Forecasting high-frequency stock returns: a comparison of alternative methods. *Annals of Operations Research*, 313(2):639–690, 2022.

[147] Shangkun Deng, Yingke Zhu, Xiaoru Huang, Shuangyang Duan, and Zhe Fu. High-frequency direction forecasting of the futures market using a machine-learning-based method. *Future Internet*, 14(6):180, 2022.

[148] Vasilios Plakandaras, Theophilos Papadimitriou, and Periklis Gogas. Forecasting daily and monthly exchange rates with machine learning techniques. *Journal of Forecasting*, 34(7):560–573, 2015.

[149] Derya Birant and Zerrin Işık. Stock market forecasting using machine learning models. In *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 1–6. IEEE, 2019.

[150] Riccardo Poli, William B. Langdon, and Nicholas Freitag McPhee. *A field guide to genetic programming*. Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`, 2008. (With contributions by J. R. Koza).

[151] Silvio Barra, Salvatore Mario Carta, Andrea Corriga, Alessandro Sebastian Podda, and Diego Reforgiato Recupero. Deep learning and time series-to-image encoding for financial forecasting. *IEEE/CAA Journal of Automatica Sinica*, 7(3):683–692, 2020.

[152] Michael Kampouridis and Edward Tsang. Investment opportunities forecasting: Extending the grammar of a gp-based tool. *International Journal of Computational Intelligence Systems*, 5(3):530–541, 2012.

[153] Chinthakunta Manjunath, Balamurugan Marimuthu, and Bikramaditya Ghosh. Deep learning for stock market index price movement forecasting using improved technical analysis. *International Journal of Intelligent Engineering & Systems*, 14(5), 2021.

[154] Partho Protim Dey, Nadia Nahar, and BM Hossain. Forecasting stock market trend using machine learning algorithms with technical indicators. *International Journal of Information Technology and Computer Science*, 12(3):32–38, 2020.

[155] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1 – 30, 2006.

[156] Salvador Garcia and Francisco Herrera. An extension on" statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of machine learning research*, 9(12), 2008.