TÉCNICO LISBOA

Machine Learning-Based Pairs Trading Strategy with Multivariate Pairs Formed with Multi-objective Optimization

Miguel Carvalho Figueira

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor: Prof. Nuno Cavaco Gomes Horta

Examination Committee

Chairperson: Prof. Pedro Filipe Zeferino Aidos Tomás
Supervisor: Prof. Nuno Cavaco Gomes Horta
Member of the Committee: Prof. Ricardo Miguel Ferreira Martins

November 2022

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

Acknowledgments

I would like to thank my supervisor, Prof. Nuno Horta, for his invaluable advice.

This work was hosted at Instituto de Telecomunicações, funded by Fundação para a Ciência e Tecnologia– Ministério da Ciência, Tecnologia e Ensino Superior (FCT/MCTES) through national funds and, when applicable co-funded European Union (EU) funds under the project UIDB/50008/2020.

I am extremely grateful to my parents and my sister, Sara, who I greatly admire, for always supporting and nurturing me.

I would like to extend my sincere gratitude to my family and friends for their encouragement and patience during this work, even when it meant spending less time with them.

Lastly, I would like to thank Marta, without whom this work would not have been possible.

Abstract

Pairs trading is one of the most popular arbitrage investment strategies. By monitoring a pair of two assets that closely follow each other, the trader acts when the pair presents an imbalance, profiting when the stocks converge to their equilibrium. Despite the rising popularity of Machine Learning in financial applications, most pairs trading strategies are still based in exhaustive search methods and rigid trading heuristics. This work creates pairs by addressing conflicting objectives, maximizing profit and minimizing risk, and explores multivariate pairs, composed of more than 2 stocks. Two Elitist genetic algorithms, NSGA II and III, are used to create pairs, achieving returns of up to 9,7% p.a., and proving to be robust to market crashes. Furthermore, multivariate pairs are found to be inferior to the traditional two-stock pairs. Additionally, a forecasting-based trading strategy is developed, attempting to improve the standard trading technique. An ARIMA and XGBoost models are used to forecast the spread in a trend-based trading strategy. The forecasting strategy yields returns of 23% p.a. and beats the market during 2018 and 2019. The system is tested in the Real Estate sector of the S&P500, from 2018 to December 2021.

Keywords

Pairs Trading, Statistical Arbitrage, Machine Learning, Genetic Algorithms, Multi-objective Optimization

Resumo

Pairs Trading é uma das mais populares estratégias de arbitragem. Através da monitorização de dois ativos que exibem uma relação de proximidade, o investidor age quando o par apresenta um desequilíbrio, lucrando quando as ações voltam a convergir. Apesar da crescente popularidade do uso de Aprendizagem Automática em áreas financeiras, a maioria das estratégias de *pairs trading* ainda se fundamenta em processos exaustivos de formação de pares e estratégias de investimento baseadas em heurísticas rígidas. Este trabalho forma pares satisfazendo objetivos contraditórios, maximizando o lucro e minimizando o risco e explora pares multi-variados compostos por mais do que duas ações. Dois algoritmos genéticos elitistas, NSGA II e III, são usados para formar pares, obtendo retornos de 9,7% por ano, e demonstrando robustez face a quedas bruscas do mercado. Adicionalmente, pares multi-variados revelam ser menos lucrativos do que os pares tradicionais de duas ações.

Complementarmente, é formulada uma estratégia de investimento baseada na previsão do comportamento dos pares, de forma a melhorar a estratégia tradicional. Um modelo ARIMA e um XGBoost são usados para prever o comportamento do par, numa estratégia baseada na tendência do preço das ações, alcançando um rendimento de 23% anualmente, superando o mercado em 2018 e 2019. O sistema é testado com ações do setor imobiliário do S&P500, de 2018 a dezembro de 2021.

Palavras Chave

Pairs Trading, Mercado de Ações, Aprendizagem Automática, Algoritmos Genéticos, Otimização Multi-objetivo

Contents

1 Introduction **1**

1.1 Origin and definition . . . 3

1.2 Exemplifying a Pairs Trading strategy execution . . . 3

1.3 The declining profitability of PT . . . 5

1.4 Thesis Objectives . . . 5

1.5 Document Outline . . . 6

2 Background and Literature Review **7**

2.1 Fundamental Concepts . . . 9

2.1.1 Cointegration - avoiding coincidences . . . 9

2.1.2 Half-life . . . 10

2.2 PT strategies classification . . . 11

2.3 Pair Formation . . . 11

2.3.1 Distance - the baseline method . . . 12

2.3.2 Cointegration . . . 13

2.3.3 Other approaches - ML and Optimization . . . 14

2.4 Trading Stage . . . 15

2.4.1 Threshold-based approach . . . 15

2.4.2 Other methods - Time series and Stochastic Control . . . 16

2.4.3 ML - Forecasting the spread . . . 16

2.5 Summary . . . 18

2.6 Benchmark Results . . . 19

2.7 Conclusion . . . 19

3 Proposed Pair Formation module **21**

3.1 Problem Statement . . . 23

3.1.1 Multivariate Pairs . . . 23

3.1.2 Multi-objective Optimization . . . 23

3.1.3 Case Studies . . . 24

3.2 Proposed Framework - NSGA II and NSGA III . . . 25
3.3 Problem Formulation . . . 26
3.3.1 Objective functions . . . 27
3.4 NSGA II . . . 28
3.4.1 Overview . . . 28
3.4.2 Main loop . . . 29
3.4.3 Non-Dominated Sorting . . . 30
3.4.4 Crowding Distance . . . 31
3.4.5 Offspring Generation . . . 32
3.5 NSGA III - Many-objectives and the ever-growing solution space . . . 33
3.5.1 Reference Points - Das and Dennis's vs Energy-based method . . . 33
3.5.2 Diversity-Preserving Operation . . . 35
3.6 Conclusion . . . 35

4 Proposed Trading Module **37**

4.1 Forecasting-Based Trading Stage . . . 39
4.2 Forecasting TS architecture . . . 40
4.3 ARIMA . . . 41
4.3.1 AR – Auto-Regression . . . 41
4.3.2 MA – Moving Average . . . 41
4.3.3 ARIMA - Adding Integration . . . 42
4.4 XGBoost - Extreme Gradient Boosting . . . 42
4.4.1 XGBoost Overview . . . 42
4.4.2 Formal definition . . . 43
4.4.3 Regularization . . . 45
4.5 Conclusion . . . 46

5 Experiment Design **47**

5.1 Data set - S&P 500 Real Estate sector . . . 49
5.1.1 Data partition . . . 49
5.2 System Architecture . . . 50
5.3 Pair Formation Stage . . . 51
5.3.1 NSGA II and NSGA III . . . 51
5.3.2 Threshold-based Trading Strategy . . . 52
5.4 Forecasting-based Trading Stage . . . 52
5.4.1 ARIMA - hyperparameter selection . . . 52
5.4.2 XGBoost - hyperparameter selection . . . 53

5.4.3 Train and Test splits . . . 54
5.4.4 Why predict the spread and not the stocks individually? . . . 55
5.5 Trading Simulation . . . 55
5.5.1 Transaction Costs . . . 56
5.5.2 Time series normalization and Look Ahead Bias . . . 56
5.5.3 Trading Portfolios . . . 57
5.6 Evaluation Metrics . . . 58
5.6.1 Forecasting Performance . . . 58
5.6.2 Trading Performance . . . 59
5.7 Conclusion . . . 59
6 Results **61**
6.1 Pair Formation Results . . . 63
6.1.1 NSGA II application . . . 63
6.1.2 Trading Performance . . . 66
6.1.3 How does a trade result in a loss? . . . 66
6.1.4 Worst performing portfolios . . . 68
6.1.5 Best performing portfolio - BOUV . . . 69
6.2 Forecasting-based Trading Stage . . . 70
6.2.1 Forecasting performance . . . 70
6.3 Forecasting trading results . . . 72
6.3.1 Why does the ARIMA portfolio perform so poorly? . . . 74
6.3.2 XGBoost: best overall portfolio - Comparison with related works . . . 75
7 Conclusions and Future Work **77**
7.1 Conclusions . . . 79
7.2 Future Work . . . 80
Bibliography **81**
A Key Performance Indicators **85**
B Parameters selection **87**
B.1 Why predict the spread directly? . . . 88
B.2 Normalization Period - 63 days . . . 88
B.3 NSGA Termination Criterion - 80 generations . . . 89

List of Figures

1.1 Example pair, its spread and corresponding strategy. . . . 4
1.2 Behavior of INTC and AAPL from 2017-2021. . . . 5
1.3 Research questions to be answered. . . . 6

2.1 Example of spurious correlation. Source [10]. . . . 9
2.2 a) Univariate, b) Quasi-Multivariate and c) Multivariate strategies. . . . 12
2.3 Validation of a Pair after OPTICS in [24]. . . . 14
2.4 Return forecast and ranking with ANN and ELECTRE III in [25]. . . . 16

3.1 Example of a Pareto-optimal front. . . . 24
3.2 Case studies for different number of stocks and objective functions. . . . 25
3.3 Pair Formation module architecture. . . . 26
3.4 Example chromosome for a universe of three stocks. . . . 27
3.5 NSGA II algorithm. . . . 29
3.6 Example solutions (a) and corresponding non-dominated fronts (b). . . . 30
3.7 Crowding Distance measure. . . . 32
3.8 Das and Dennis's reference points. Adapted from [37]. . . . 34

4.1 Trend-based forecasting strategy. . . . 40
4.2 Forecasting TS module architecture. . . . 41

5.1 Sliding window periods. . . . 50
5.2 System architecture. . . . 50
5.3 Trend and seasonal components of FRT-EQR in 2019. . . . 53
5.4 Evolution of the *rmse* metric for the spread DLR-AVB . . . 54
5.5 XGBoost forecasting of KIM-ESS for different N. . . . 55
5.6 Comparison of the normalized ESS-DLR spread using past and future values. . . . 57
5.7 Portfolio management during the experiment. . . . 58

6.1 Number of stocks per component in BOMV (a) and MOMV (b) pairs. 64
6.2 Constraint violation (a), evolution of the objective functions (b) and selected pairs (c) for BOMV pairs in 2018. 64
6.3 Example of pair (a) and spread (b) that result in a loss. 67
6.4 Different behavior of EQR-DLR in the training and testing periods. 69
6.5 Influence of non-convergence in BOUV (a) and SOUV (b) in 2018. 69
6.6 Comparison of SPY and BOUV portfolios in 2018-2021 (a), and during 2020 only (b). . . 70
6.7 XGBoost and ARIMA forecasts for KIM-ESS in 2018. 72
6.8 Xgboost, ARIMA and BOUV portfolios compared with the SP500 in 2018-2019. 74
6.9 ARIMA forecast for KIM-ESS in June 2018. 75

B.1 Evolution of the objective space in 2019 for the BOUV (a), BOMV (b), MOUV (C), MOMV (d) pairs. 90

List of Tables

2.1 Summary of ML techniques in PT. 18
2.2 Returns of CI strategies. 19
2.3 Returns of non CI strategies. 19

3.1 Domination count and dominated set for some example individuals. 31
3.2 Key Points from the Pair Formation (PF) framework. 36

4.1 Main characteristics of the proposed Trading Model. 46

5.1 Threshold strategy parameters. 52
5.2 Transaction costs considered. 56
5.3 Hyperparameters for the algorithms employed. 60

6.1 Number of pairs selected for the different types of pairs. 63
6.2 Trading Performance of the different types of pairs during 2018 to 2021. 65
6.3 Number of pairs still cointegrated over the total number of pairs. 68
6.4 Forecasting performance of ARIMA and XGBoost in 2018 and 2019. 71
6.5 Forecasting-based strategy performance in 2018 and 2019. 73
6.6 Exemplifying the forecasting strategy for ARIMA. 74

B.1 Comparison of predicting the spread directly or the 2 stocks individually 88
B.2 Trading results for different normalization periods from 2018 to 2021. 89

Acronyms

ADF	Augmented Dickey-Fuller
ARIMA	Autoregressive Integrated Moving Average
BOMV	Bi-Objective Multivariate
BOUV	Bi-Objective Univariate
CI	Computational Intelligence
ETF	Exchange-Traded Fund
GA	Genetic Algorithm
GGR	Gatev, Goetzmann and Rouwenhorst
LSTM	Long Short Term Memory
KPI	Key Performance Indicators
MDD	Maximum Drawdown
ML	Machine Learning
MOEA	Multi-Objective Evolutionary Algorithms
MOMV	Many Objective Multivariate
MOOP	Multi-Objective Optimization Problem
MOUV	Many Objective Univariate
NSGA	Non-dominated Sorting Genetic Algorithm
NZC	Number of Zero Crossings
PF	Pair Formation
PT	Pairs Trading
ROI	Return On Investment
RS	Research Stage

SOUV	Single Objective Univariate
SSD	Sum of Square Distances
TS	Trading Stage
XGBoost	Extreme Gradient Boosting

1

Introduction

Contents

1.1 Origin and definition 3
1.2 Exemplifying a Pairs Trading strategy execution 3
1.3 The declining profitability of PT 5
1.4 Thesis Objectives 5
1.5 Document Outline 6

1.1 Origin and definition

Pairs Trading (PT) was introduced mid 80's by a team of analysts from Morgan Stanley and is one of the speculation strategies most used by hedge funds. PT is characterized by being highly analytical, relying on statistic and technical analysis. It belongs to a set of tools known as statistical arbitrage.

The strategy is based on the idea that two securities that are close substitutes to each other, whether because they belong to the same sector or are affected by the same economic factors, shall have similar prices (Law of One Price). As such, if their prices do differ, that is due to the market inefficiency, and it is an opportunity to act as an arbitrageur and make profit, betting that their price will soon be similar again.

In its simplest form, the strategy is composed of two steps:

Pair Formation (PF):

1. Identify two securities (stocks, Exchange-Traded Fund (ETF)s, etc.) whose price series historically move together.

Trading Stage (TS):

2. If the price series diverge and, consequently, the spread widens, the pair is unbalanced. A trade is initiated, "shorting the winner and buying the loser". [1]

3. When the price series of each component converge and the spread is reduced, the trade is closed.

In a Pairs Trading strategy, the spread is meticulously analyzed and is defined as the difference between the price of each security. For some given securities, X and Y, the spread is expressed as

$$S_t = X_t - Y_t. \tag{1.1}$$

1.2 Exemplifying a Pairs Trading strategy execution

A simplistic example of how the strategy is applied is presented next.

In fig. 1.1 it is possible to observe the evolution of the price series of BXP and WELL, two stocks from the real estate sector. The price series seem to move together. As so, BXP and WELL will constitute a pair in this example. The pair's spread, given by $S_t = BXP_t - WELL_t$, will be monitored closely to detect unbalances in the pair's behavior that could translate to trading opportunities.

Two thresholds are defined, which specify the minimum required deviation to open a position. When the spread reaches the "Short Threshold", its value is higher than anticipated, and it is presumed that it will decrease. Therefore, the spread is shorted by buying WELL and shorting BXP. The opposite occurs for the "Long Threshold". The positions are closed when the pair converges, and the spread reverts to its expected value. The spread, the thresholds and the current strategy position can be observed in the

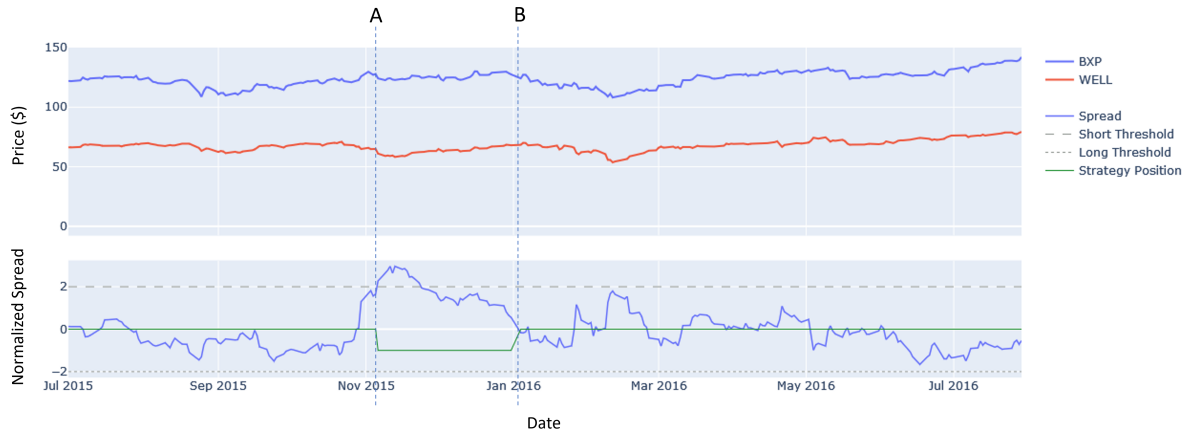


Figure 1.1: Example pair, its spread and corresponding strategy.

bottom subplot of fig. 1.1. The strategy position assumes the values -1, 0, or 1, depending on whether the current position is short, closed, or long, respectively.

In instant A, around November of 2015, the securities' prices exhibit different behaviors. In particular, WELL devalues more than BXP. As the spread hits the Short Threshold, a short position is entered. If the pair continues to act as it has in the past, it is expected that the pair's components will converge. Based on this assumption, one will bet that spread's value will decrease - short the spread. Effectively, in instant B, the pair converges and the position is closed with profit.

This strategy, although disarmingly simple, provides positive returns as demonstrated in [1], where the strategy described in the example is first formulated. When applied to the S&P500 from 1962 to 2002, returns up to 11% per annum were obtained.

Pairs Trading is characterized by being market neutral, meaning that it attempts to avoid systematic risk and be profitable both in bullish and bearish markets. It also has low correlation with the market, corroborating that it is market neutral, as inferred in [2].

PT's ability to make profit arises from market inefficiencies, [3], such as trades from uninformed buyers or delays in information diffusion across the two components of a pair [4]. It is also stated in [5] that PT's performance is the highest in low information markets. Consequently, it is possible to conclude that the market structure influences the strategy's returns, which is verified by [6], that compares the technique on the Shanghai, Shenzhen and Hong Kong stock markets.

One final advantageous feature consists in focusing on the relative pricing of a security, bypassing the inconvenience of evaluating an asset.

1.3 The declining profitability of PT

However, it seldom happens that the market behaves as one would desire. It may be that a ”well behaved pair” suddenly goes rogue. In other words, following the opening of a trade, the pair further diverges.

Let's assume that Intel (INTC) and Apple (AAPL) are co-moving and form a pair. From June 2017 to March 2020, the price series followed each other closely, as presented in fig. 1.2. Nevertheless, from March onward, the stocks diverged greatly. An unprepared trader would open a trade when they first separated and suffer great losses, since the short on AAPL would devalue tremendously.

Figure 1.2: Behavior of INTC and AAPL from 2017-2021.

In fact, although PT performed well through the 80's and 90's, its profitability has been decreasing. In [1] it is hypothesized that the influence of hedge funds was the main contributor to PT's declining performance, as their considerable buying and computing power is able to quickly act upon interesting pairs, leaving no margin to the common retail trader. However, [7] states that the diminished returns are driven by worse arbitraging risks, namely the fundamental risk (associated to the market as a whole), or noise-trader risk (attributed to irrational choices by traders). [7] concludes that a naive method of pairing securities, as the one introduced above, will not be enough to generate consistent returns.

The example presented in fig. 1.2 and the motives behind PT's poor performance highlight the importance of choosing quality pairs and the need for a robust trading system, capable of dealing with ”bad behaved” pairs.

1.4 Thesis Objectives

Computational Intelligence (CI) has influenced greatly the financial industry. From the sheer volume of capital traded daily, to the innumerable high frequency transactions attributed to trading algorithms, the market has become even more dynamic and competitive. In [8], the author compares econometric models with Machine Learning (ML) methods and the latter largely outperforms the former when handling

large amounts of data. In contrast, Pairs Trading techniques are still largely based in ”human” heuristics, representing an interesting research opportunity.

The success of a PT strategy relies mostly on a good selection of pairs, but the traditional iterative approaches, that compare each stock with all others, consume immense computational power and do not guarantee that the relationships found are not coincidental. Additionally, PT, as most financial strategies, tries to conciliate two opposite goals: minimize risk and maximize profit. To address this dichotomy, in this work, the problem of PF will be formulated as a Multi-Objective Optimization Problem (MOOP), and Genetic Algorithms will be used to form the best possible pairs. Furthermore, pairs composed of more than the standard two securities will be formed, as the multivariate kind of PT is still unexplored and could produce better results than its univariate equivalent.

Selecting the right pairs is crucial but knowing when to trade them is just as important. To this end, the vast arsenal of ML will be employed to forecast the spread of each pair and generate trading signals. In the forecasting-based TS module, an Autoregressive Integrated Moving Average (ARIMA) and Extreme Gradient Boosting (XGBoost) models will be used to anticipate trading opportunities.

The contributions that this thesis pretends to add to the literature can be synthesized by the questions presented in fig. 1.3. Research Stage (RS) I will address the PF phase, and RS II the TS.

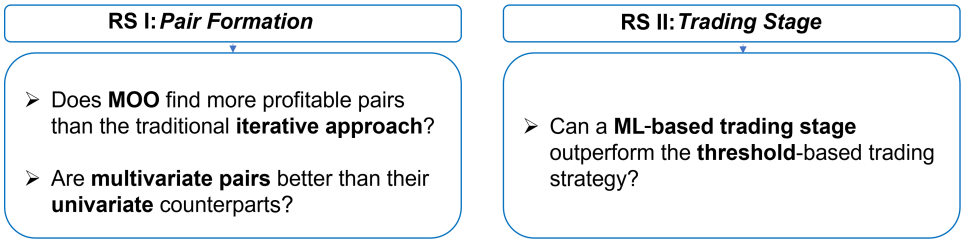


Figure 1.3: Research questions to be answered.

1.5 Document Outline

This work has a total of seven chapters. Chapter 2 comprises a review of the Pairs Trading literature, its background and the state-of-the-art techniques. Chapter 3 and 4 describe the framework developed to answer the research questions formulated. In particular, chapter 3 describes a novel method to group pairs, concerning the first research stage, RS I. Chapter 4 delineates a forecasting-based TS, using ML models to predict the spread, addressing the RS II. Chapter 5 details the experiment design and highlights some features of the back-testing simulation that ensure the robustness of the algorithms in a live trading environment. The results obtained are analyzed in chapter 6. Finally, chapter 7 summarizes the findings and contributions of this work and answers the questions that motivated this thesis.

2

Background and Literature Review

Contents

2.1 Fundamental Concepts 9
2.2 PT strategies classification 11
2.3 Pair Formation 11
2.4 Trading Stage 15
2.5 Summary 18
2.6 Benchmark Results 19
2.7 Conclusion 19

In this chapter, the fundamental concepts of Pairs Trading are presented, and an assessment of the existing literature is made.

2.1 Fundamental Concepts

Before examining the Pairs Trading strategy and the different existing techniques, the fundamental underlying notions that define the relationship between two price series must be studied. In fact, the soundness of a pair relationship is crucial to the success of this investment strategy, as emphasized in section 1.3.

2.1.1 Cointegration - avoiding coincidences

Before the introduction of cointegration tests, economists relied on linear regression to find the relationship between several time series processes. The concept of spurious correlation was first introduced in 1920, denoting a mathematical relationship in which two variables are associated but not causally related [9]. In other words, the relationship observed between two variables is merely coincidental. Fig. 2.1 displays a curious correspondence between two events whose relation is surely accidental, although they seem to be correlated.

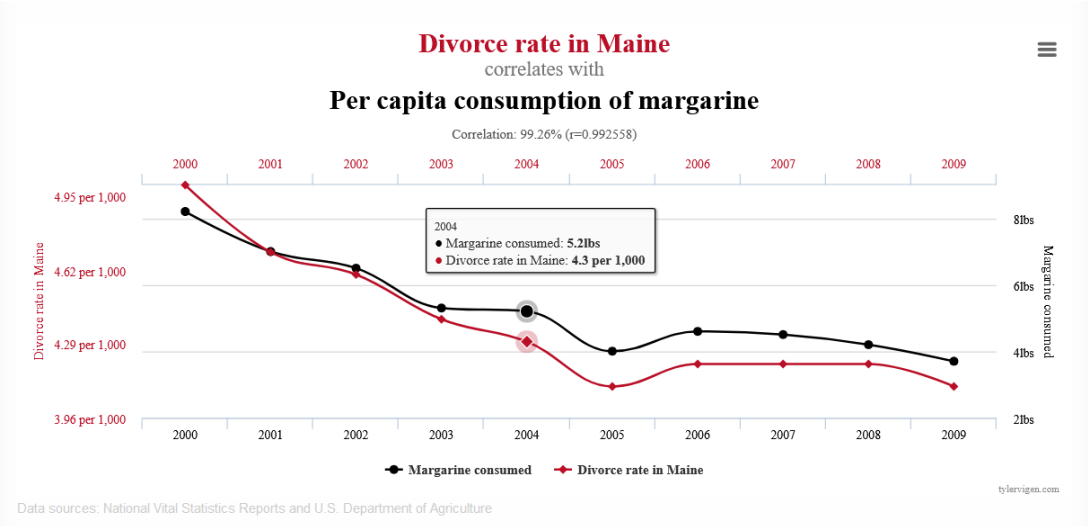


Figure 2.1: Example of spurious correlation. Source [10].

Introduced in 1987, by the Nobel laureates Engle and Granger, the concept of cointegration establishes a trustier long-term association between two time series, avoiding the danger of assuming fortuitous relationships. Formally, it can be described as in def. 2.1.2. The concept of order of integration is defined in def. 2.1.1 [11].

Definition 2.1.1. A series x_t which has a stationary representation after differencing *d* times, is said to be integrated of order *d*, denoted $x_t \sim I(d)$.

Definition 2.1.2. Two given series x_t, y_t, each $I(d)$, are said to be cointegrated if there is a linear combination $z_t = x_t - \beta y_t$ so that $z_t \sim I(d-1)$. The parameter β is called the cointegration factor.

In the case that the order of integration, *d*, is equal to 1, the def. 2.1.2 could be translated to: "if two price series are cointegrated, their spread, given by $s_t = x_t - \beta y_t$, is I(0), i.e. is stationary. This is an important conclusion since it presents an opportunity to artificially derive a stationary time series.

A stationary time series is particularly interesting in a PT strategy, since it inherently is mean reverting. As consequence, a trader can analyze a stationary spread to detect instants where it deviates from its habitual behavior, thus finding trading opportunities. Revisiting fig. 1.1, it is possible to observe that the normalized spread is indeed stationary and habits mainly close to its mean.

The most commonly employed method for testing for cointegration, the Engle-Granger Test, starts by creating residuals based on a least squares regression. Using the Augmented Dickey-Fuller (ADF), the residuals are then inspected for the presence of unit-roots. If the hypothesis of the existence of said unit roots is rejected, the residuals are stationary, and the price series are cointegrated. In conclusion, if the time series are cointegrated, the Engle-Granger method will demonstrate the stationarity of the spread.

2.1.2 Half-life

Chan [12] proposes a method to calculate the speed of mean reversion, denominated half-life. Drawing inspiration from the field of physics, where half-life denotes the time needed for a given substance to decay to half its mass, the half-life expresses the time that a series takes to revert to its mean.

The author describes a time series as an Ornstein Uhlenbeck Process,

$$dy(t) = (\lambda y(t-1) + \mu)dt + d\epsilon, \tag{2.1}$$

where $d\epsilon$ is some Gaussian error, μ is a constant, and coefficient λ establishes the relationship between $y(t)$ and $y(t-1)$.

This formulation makes it possible to derive an analytical solution for the expected value of y(t),

$$E(y(t)) = y_0 \exp(\lambda t) - \frac{\mu}{\lambda(1 - \exp(\lambda t)}. \tag{2.2}$$

In a PT context, describing the spread as an Ornstein Uhlenbeck Process has multiple advantages. Firstly, if for a given spread the coefficient λ is found to be negative, the spread is proven to be mean reverting, which is desirable. Secondly, it allows an estimation of the time that it takes to revert to the mean, the half-life. Indeed, if $\lambda < 0$, the half-life decays with $-log(2)/\lambda$. Naturally, the shortest the

half-life, the higher the profit.

Having describe the fundamental concepts of Pairs Trading, the following section reviews the existing strategies found in the literature.

2.2 PT strategies classification

A Pairs Trading strategy can be divided into its two logical steps: Pair Formation (PF) and Trading Stage (TS). In the PF, co-moving securities are grouped into pairs. Afterwards, in the TS, the spread of the selected pairs is kept under analysis and the pair is traded according to its spread's behavior. In the literature, there are several methods regarding the clustering of securities, whether based on correlation, cointegration, among others. Similarly, the authors studied propose contrasting trading techniques, that differ when choosing when to trade the pair.

Following the comprehensive survey presented in [4] and [13], Pairs Trading approaches can be classified as:

- Distance, Cointegration, Other (Machine Learning, Optimization, etc.).
- Thresholds, Time Series and Stochastic Control, Machine Learning.

The first techniques concern the Pair Formation problem, and the second the Trading Stage.

2.3 Pair Formation

The Pair Formation problem involves pairing securities in a certain set, followed by choosing the pairs with the highest potential.

A standard pairing technique consists of grouping securities belonging to the same sector, with the intent of ensuring that they are influenced by the same external factors. This procedure can be observed in [1, 7, 14]. In [15], where pairs are formed in Taiwan's semiconductor market, the authors conclude that pairing stocks among the same sector outperforms the case where groups are formed without this restriction.

On the other hand, an extensively iterative approach is used in [16, 17], pairing all combinations of securities without constraints concerning the sector.

PF methodologies can be further categorized with respect to the number of assets that constitute each component of the pair. Fig. 2.2 illustrates the various approaches present in the literature.

A univariate pair is the most commonly employed, where a security is coupled with a single other (fig. 2.2 a)). In a quasi-multivariate scheme, a stock is matched with several other (fig. 2.2 b)). Finally, a multivariate framework groups a set of securities with other set. Each set is obviously composed of

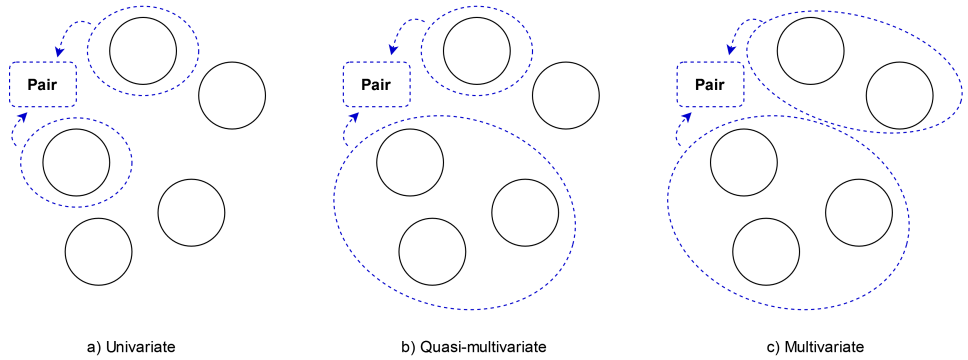


Figure 2.2: a) Univariate, b) Quasi-Multivariate and c) Multivariate strategies.

various assets (fig. 2.2 c)). Although these procedures vary in the composition of a pair, the Trading Stage would be the same for every one of them: each leg of the pair has a price series, and the distance between the legs' price series is the spread, that will be scrutinized continuously. The only difference consists on how the price series is obtained: in a quasi-multivariate or multivariate strategy the component's price series is the average of each of its constituents' series.

The majority of the academic works on PT explores the univariate method, with rare exceptions, namely [5, 15, 18], who implement a quasi-multivariate strategy or [19], where a truly multivariate framework is tested. The texts cited above will be analyzed in further detail, but the reduced number of works that delve into the multivariate kind reveals an apparent absence of this alternative methodology.

Having formed the pairs, it is now time to select the best ones using Distance, Cointegration or ML criteria.

2.3.1 Distance - the baseline method

In one of the first academic works about Pairs Trading, Gatev et al. [1] use the euclidean distance to evaluate pairs. The ones that present the smallest Sum of Square Distances (SSD) during a rolling window of 12 months are selected. The authors resort to this criterion due to its simplicity and transparency, making it easy to apply to large data sets and robust against data snooping. This technique is commonly referred as Gatev, Goetzmann and Rouwenhorst (GGR), in honor of its authors.

The SSD is also employed in [18], where a stock is matched with a set of 5 other that minimize the SSD. The stock is then traded in quasi-multivariate fashion against the assembled set, as it will be described in the trading stage section.

The SSD criterion, despite being compellingly simple, is naive and analytically sub-optimal. A rational trader would desire a pair that diverges often and proceeds to converge, since it would provide a high number of trading opportunities, i.e., a pair whose spread has high variance and is strongly mean-reverting. The SSD, however, would promote a less desirable "stable" pair instead of the former. An

even stronger flaw could be pointed out: in a worst case scenario where no quality pairs exist, the pair with the minimum SSD is still a bad pair. Concluding, this approach does not consider the case where there are no good pairs.

An improvement to the SSD is suggested in [7]. In the article by Do et al., GGR's method is tested and its declining profitability is interpreted. The authors propose the use of the Number of Zero Crossings (NZC) by the pair's spread as a measure of mean regression. An appealing pair should have a higher NZC.

2.3.2 Cointegration

The cointegration approach tests the pair's spread for stationarity, namely with the ADF, the Engle-Granger or the Johansen tests. The theory in which this approach is rooted dates back to [20, 21], naturally.

According to [4], the key benefit of this strategy is the econometrically trustier association of identified pairs, opposing the distance method, which may find spurious relationships.

Vidiamurthy [22] is the reference when it comes to univariate pairs trading using cointegration. The price of a security, p, is decomposed in its stationary (s) and common trend (c) components:

$$p = s + c \tag{2.3}$$

$$\Delta p = r = r^c + r^s \tag{2.4}$$

Like so, its return, r, comprises a common trend return, r^c, and a stationary specific return, r^s, as seen in equation (2.4).

Remembering that the spread, s, is the difference between the price series of each stock, for stocks i, j its variation is given by:

$$\Delta s = \Delta p_i - \Delta p_j = r_i - r_j \tag{2.5}$$

Vidiamurthy concludes that, for a pair to be cointegrated, the common return components, $r^c_{i,j}$, should be identical up to a scalar, the cointegration coefficient. The r^c would then cancel out, leaving the specific return of each stock, $r^s_{i,j}$, which is stationary as previously mentioned.

Caldeira and Moura [23] compare the use of correlation and cointegration when forming pairs. Correlation commonly refers to the degree to which a pair of variables are linearly related and reveals short term dependencies. As such, the authors state that the use of cointegration is more adequate to a PT strategy, as it expresses long term relations. The pairs are formed between stocks belonging to the

Ibovespa and are tested for stationarity with the ADF and the Engle-Granger tests. Returns up to 16% per year are obtained. It is also proposed a cashless strategy, where the capital obtained from shorting a stock is applied to its counterpart long component. This way, the trader would not need to invest their own money.

2.3.3 Other approaches - ML and Optimization

Given the advent of computational power and the broad applications of ML in finance, one would believe that its use would be customary when clustering securities. It is not the case, though. The instances where computational intelligence is applied to the Pair Formation problem are rare.

Sarmento [24] applies the OPTICS algorithm to pair stocks within a set without the need to specify the number of clusters. Allied with the use of Principal Component Analysis and Hurt's exponent, OPTICS addresses varying cluster densities and facilitates the investor's task. This unsupervised learning algorithm achieved better risk-adjusted returns than searching within a sector or iterating over all possible pairs. Fig. 2.3 shows the validation process of a pair clustered by the OPTICS algorithm. The author imposes that the Hurst exponent, H, the half-life, hl and the number of times the spread crosses its mean, *mean cross*, must obey certain constraints. The half-life expresses how much time a pair takes to converge, and it is an useful parameter to monitor since it is conjectured that pairs that do converge tend to do it in the first week after the initial divergence [17].

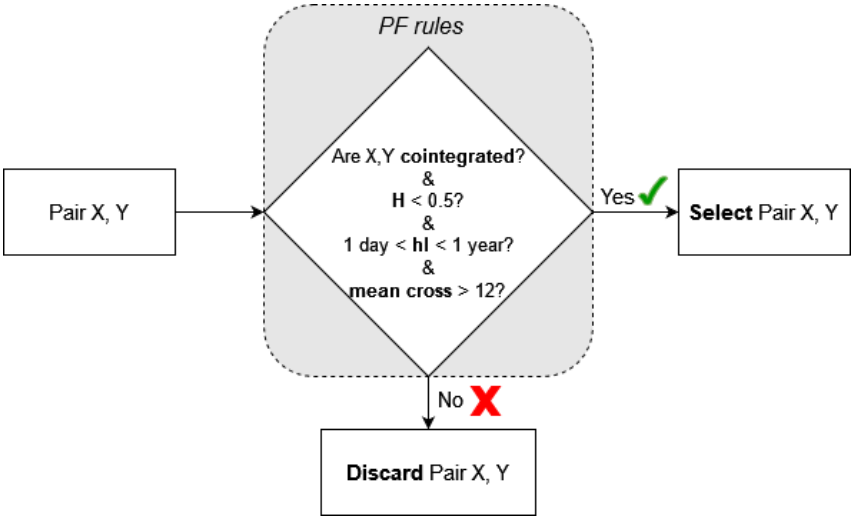


Figure 2.3: Validation of a Pair after OPTICS in [24].

Goldkamp et al. [19] introduce a multivariate and multi-objective technique by formulating the pairing stage as an optimization problem. This work is groundbreaking in a sense that it conciliates two disparate objectives: maximizing profit and minimizing risk. It is also the only work found that explores multivariate pair formation with the use of computational intelligence.

The authors employ Non-dominated Sorting Genetic Algorithm (NSGA) II to find pareto fronts. There are no constraints regarding the sector, different types of stocks can constitute the same component of the pair. The method is tested in the S&P500, beating the market and achieving a low β. In other words, it proves that PT has low correlation with the market.

2.4 Trading Stage

Following the formation of the pairs, the Trading Stage encompasses deciding when to open and close a trade. Most of the strategies, if not all of them, rely on analyzing the pair's spread and making decisions according to its behavior.

It must be noted that the typical procedure is to apply the same amount of capital to each component of the pair. To rephrase it, the value obtained in the short is equal to the value spent on the long.

Two main approaches will be reviewed: the traditional Threshold-based approach and CI techniques, including ML and Optimization[1].

2.4.1 Threshold-based approach

In the work of Gatev et al. [1], the distance method of pairing securities is followed by an even simpler trading model. The pair's spread is monitored and, when it diverges by more than two times its standard deviation, the trade is opened. When the spread converges to zero, the trade is closed. This methodology is the one exemplified in the Introduction.

Other notable works already mentioned that follow this technique are [18, 19].

Perlin [18] explores a quasi-multivariate framework, coupling a single stock, i with 5 other that are co-moving. The price series i is then expressed by a linear combination of the price of the other 5 assets and an error term (ϵ).

$$p_i = \sum_{j=1}^{5} w_j P_j + \epsilon \tag{2.6}$$

The weights (w) are re-estimated every ten-days using the past two years of observations.

However, in Perlin's experiment only the singular stock is traded, and not the cluster of 5 stocks.

In its multivariate formulation, Goldkamp et al. [19] only address the problem of pair formation, opting for a simplified trading scheme using thresholds. The authors state that future work should implement an intelligent trading strategy.

[1] For simplicity's sake, Optimization, in particular GAs, will be referred under the umbrella of ML.

2.4.2 Other methods - Time series and Stochastic Control

In both these approaches, the pair formation period is generally ignored, as the authors assume that co-moving securities have already been grouped. Instead, the main focus of these domains is the trading period.

The Time Series method examines how to optimize trading signals by applying different methods of time series analysis.

The Stochastic Control approach studies the optimal portfolio holdings in each component of the pair. Naturally, it draws from stochastic control theory.

2.4.3 ML - Forecasting the spread

Machine Learning is suited to generate predictions. Given the huge quantity of financial information available, only computational methods can deal with such amount of data. As expected, the applications of ML to predict economic trends are becoming more and more common. In fact, several authors employ ML algorithms to forecast the behavior of a pair's spread. As such, they will be able to anticipate trading opportunities.

- Artificial Neural Network (ANN)

Huck [25, 26] uses an ANN to forecast a stock's growth. The best pairs are then selected with the help of a multi-criteria decision-making algorithm, ELECTRE III. This algorithm is especially useful to rank actions by order of preference. In other words, the stocks will be ranked according to their potential growth. Finally, the stocks on the top will be bought and the bottom ones shorted. Note that this strategy differs from traditional pairs trading in a sense that it does not group stocks that are co-moving, but rather opposite stocks (the ones with highest potential with the ones with the least). This method is also known as "Top-Bottom Strategy". Fig. 2.4 illustrates Huck's strategy.

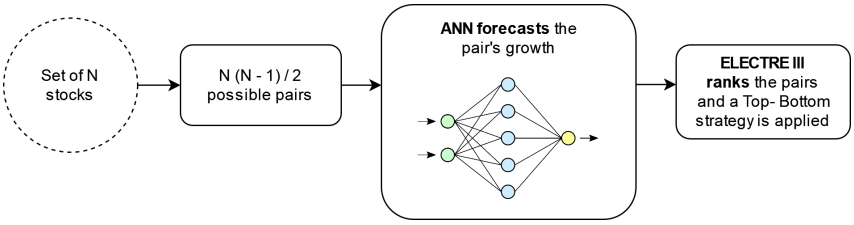


Figure 2.4: Return forecast and ranking with ANN and ELECTRE III in [25].

- Reinforcement Learning (RL)

Both [16, 17, 27] base their approach on a RL algorithm.

Wang [17] and Kim et al. [27] implement a Deep Q-Network (DQN) as the trading agent, after forming the pairs with the Engle-Granger test.

DQN is used as the RL agent. But, instead of treating the problem as a black box which rewards the strategy with the highest return, a reward shaping method is implemented. The reward (R) is defined as

$$R = r - \beta \cdot |a_{RL} - \pi_B|. \tag{2.7}$$

Simply put, a penalization is subtracted from the return (r) if the action chosen by the RL agent (a_{RL}) differs from a baseline strategy (π_B). This baseline strategy is a "human" conceived strategy.

To summarize, this method conciliates already made "human" strategies with a state-of-the-art RL approach. The authors conclude that the model outperforms GGR's method and that the most profit is made in the first days after a trade is initiated. This would lead to believe that or a pair converges quickly, or potential profit will be erased by the short's premium.

Brim [16] applies a similar technique, comparing a regular DQN with a Double DQN (DDQN), as the one used by DeepMind's AlphaGo. The author reaches the conclusion that the DDQN makes much fewer predictions than the DQN, thus resulting in worse but less volatile returns.

- Genetic Algorithm (GA)

Huang et al. [15] resort to a GA to optimize their trading strategy. Ten companies in the Taiwanese semiconductor market are analyzed, and Bollinger Bands are employed to detect if a stock is diverging from the group. As a matter of fact, the authors unknowingly implement a quasi-multivariate strategy, since their method trades a single stock paired to a cluster of several other. The GA will define the period n relevant to the moving average, the Bollinger Band's parameters and how much to buy of each stock.

The authors then repeat the experiment with a group of the 10 companies with the highest market cap, once again in the Taiwanese market. The same sector case outperforms the heterogeneous one, and it is inferred that Pairs Trading within the same sector yields better returns.

- Long Short Term Memory (LSTM)

LSTM algorithms are put to use in [14].

Zhang [14] aggregates stocks from the Chinese automotive sector, testing the pairs for cointegration with the Engle-Granger test. Secondly, a LSTM predicts the spread's evolution. The author justifies the use of a LSTM stating that this algorithm keeps memory of the long past opposing, for example, a RNN that is only influenced by recent data.

2.5 Summary

Table (2.1) encapsulates ML techniques applied to a PT strategy. In the *Methodology* column, (PF) designates that the work refers to the Pair Formation problem, and (TS) that it alludes to the Trading Stage. Additionally, it is included [28], where Extreme Learning Machine (ELR) and Support Vector Regression (SVR) algorithms are combined with a Kalman Filter (KF) to better model the spread.

Table 2.1: Summary of ML techniques in PT.

Work	Methodology	Data	Conclusions
Sarmento [24]	(PF) OPTICS applied to the PF problem.	Commodity linked ETFs 2009-2019	OPTICS achieved better risk-adjusted returns than searching within a sector or iterating over all possible pairs.
Goldkamp et al. [19]	(PF) NSGA II clusters pairs in multivariate fashion.	S&P 500 2012-2016	Multi-objective formulation outperforms single objective. Multivariate PT has low β.
Brim [16]	(TS) DDQN vs DQN	S&P 500 2014-2017	DDQN is more conservative and produces less volatile returns.
Wang et al. [17]	(TS) RL with DQN	NASDAQ Nordic Exchange 2009 -2013	RL outperforms GGR's method. Most profitable trades do not take very long to converge.
Nobrega et al. [28]	(TS) ELM and SVR allied with a KF	BM & FBovespa (Brasil) 2009-2013	Both ELM and SVR perform better when allied with a KF.
Huang et al. [15]	(TS) GA to optimize entry points.	Taiwanese stock market 2003-2012	Trading within a sector outperforms trading all types of stocks.
Flori et al. [2]	(TS) LSTM predicts the probability of a stock outperforming its peers.	S&P 500 2000-2019	In the 2008 crisis the strategy generated consistent returns, even though volatility increased. The strategy outperformed the market greatly.
Zhang [14]	(TS) LSTM	Chinese stock market 2015-2020	In market turmoil the strategy under-performs GGR. The predictive model with a LSTM is less sensitive to sudden changes.
Huck [25, 26]	(TS) ANN associated with ELECTRE III	S&P 100 1992-2006	In a Top-Bottom strategy, the fewer the pairs the better the returns.

Summarizing, ML has been applied mostly to the Trading Stage, making use of its ability to predict price movements having past data as input. The use of computational intelligence in Pair Formation is rather rare and, considering the numerous ML algorithms that deal with clustering, this presents an opportunity for future research. It should also be noted that the multivariate technique is almost absent in the literature despite having several advantages over its univariate equivalent.

2.6 Benchmark Results

In this section, the returns obtained in several studies on the topic, namely the ones mentioned in the previous sections, will be reviewed and compiled, to ballpark results.

Given the unpredictable nature of both the market and ML performance, it would be challenging to foretell the performance of the intelligent Pairs Trading agent that will be developed during the thesis. What could be done instead is to set goals for the agent to achieve. As it might be expected, the system is required to match similar ML experiments and outperform non-computational methods, otherwise the purpose of using computational intelligence is defeated.

Table 2.2 shows the results obtained in three different works that employ a ML-based PT strategy. It should be pointed that, although all strategies produce positive returns, only [24] considers transaction costs in their approach.

Table 2.2: Returns of CI strategies.

Work	Methodology	Data	Transaction Costs	Results
Brim [16]	RL with DDQN	S&P 500 2018	No	131.33% cumulative returns
Sarmento [24]	OPTICS	Commodity linked ETFs 2009 - 2019	Yes	12.5 % ROI 3.79 Sharpe Ratio
Goldkamp et al. [19]	NSGA II	S&P 500 2012-2016	No	1.72% monthly return

Table 2.3 resumes the returns of non CI methods. It is immediately noticeable that the strategies presented in table 2.2 are superior, both in return and Sharpe Ratio.

Table 2.3: Returns of non CI strategies.

Work	Methodology	Data	Transaction Costs	Results
GGR [1]	Distance / Threshold (GGR's strategy)	S&P 500 1962-2002	Yes	11% excess returns p.a. 0.45 Sharpe Ratio
Do et al. [7]	Improved GGR	S&P 500 1962-2002	No	0.84% monthly 0.52 Sharpe Ratio (subset of Bank stocks)

2.7 Conclusion

In this chapter, the theoretical concepts of PT and the techniques present in the literature, both regarding the formation and the trading of pairs, were analyzed. The results of pertinent works were compiled into section 2.5. Finally, the trading results found in the literature, displayed in section 2.6, will be used as a benchmark to assess the relative performance of the framework developed.

3

Proposed Pair Formation module

Contents

3.1 Problem Statement 23
3.2 Proposed Framework - NSGA II and NSGA III 25
3.3 Problem Formulation 26
3.4 NSGA II 28
3.5 NSGA III - Many-objectives and the ever-growing solution space 33
3.6 Conclusion 35

In the previous chapter, the PT strategy and the corresponding existing techniques were reviewed. In this chapter, a novel Pairs Formation framework is presented, aiming to improve the PF stage and to answer the two research questions regarding the formation of pairs:

- Are **multivariate** pairs better than **univariate** ones?
- Is it advantageous to formulate the PF problem as a **MOOP**?

3.1 Problem Statement

3.1.1 Multivariate Pairs

The mantra popularized by the likes of Benjamin Graham and Warren Buffet - ”Don’t put all your eggs in one basket” – encapsulates the motive for using multivariate pairs, where each component of a pair is composed of several securities.

The motivation behind the multivariate technique relies on the idea that increasing the number of securities held reduces volatility. In [19], it is verified that not only multivariate PT produces less volatile results but also presents lower correlation with the market when compared to its univariate counterpart. Additionally, [18] and [5] both conclude that it obtains higher returns.

The combinatorial nature of multivariate PT poses, however, a problem. Given n different stocks, the number of pair combinations, P, is described by equation 3.1 where the number of securities in each component, a, ranges from 1 to b. There are 2 components in a pair, naturally.

$$P = \binom{\sum_{1}^{b} \binom{n}{a}}{2} \tag{3.1}$$

Ultimately, the number of configurations is a combination of combinations, and will grow enormously. In this work, multivariate pairs will be formed within the same sector. The benefits of this restriction are twofold: Firstly, the dimension of the set from where pairs are selected is reduced greatly, solving the combinatorial exponential growth and avoiding the need for immense computational power; Secondly, and more importantly, the securities are ensured to share a common background, since they are exposed to similar underlying economic factors. Consequently, the pairs formed will be sound and not a simple statistical coincidence. [15] corroborates this proposition and concludes that PT within a sector yields better results than a strategy without this constraint.

3.1.2 Multi-objective Optimization

Like most trading strategies, PT aims to maximize profit while minimizing risk. Nonetheless, these objectives oppose each other, inviting the implementation of a multi-objective framework.

In a MOOP the set of solutions reflects the best trade-off between competing objectives, constituting the Pareto-optimal set. The boundary defined by the set of all points mapped from the Pareto optimal set is called the Pareto-optimal front, depicted in fig. 3.1.

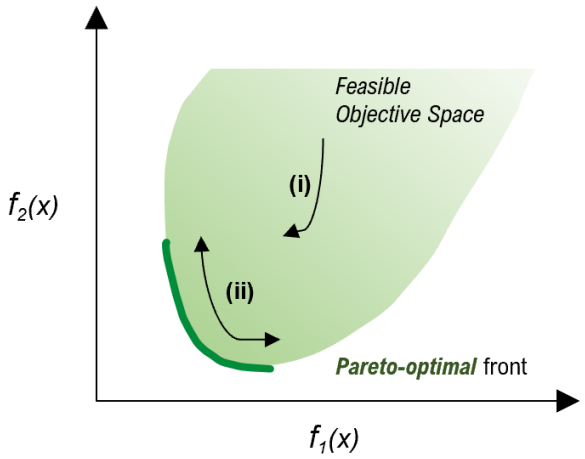


Figure 3.1: Example of a Pareto-optimal front.

The goal of MOOP is to find solutions as close (i) and well spread (ii) as possible to the Pareto front, ensuring the best and most diverse set of solutions. Fig. 3.1 exemplifies a Pareto-optimal front for two given functions and the two main objectives of a MOOP.

In this work, the MOOP will be solved with a GA. GAs present some advantages over traditional gradient-based optimization methods:

- Rather than executing a point-to-point search, they incorporate a population of solutions, thus finding a diverse set of solutions instead of a single one.
- Use random variation to explore for new solutions, instead of gradient information. This avoids stalling or being trapped in local minima.

3.1.3 Case Studies

To address the questions posed in RS I, five different types of pairs will be formed.

To test whether multivariate pairs are superior to univariate ones, univariate pairs (1 stock to 1 stock) will be compared with multivariate (several stock to several).

To examine if pair formation benefits from being formulated as a MOOP, pairs will be created by minimizing either one (single objective), two (bi-objective) or many (four; many-objective) goal functions.

As result, Bi-Objective Univariate (BOUV) pairs are created according to two objectives, and are univariate. Bi-Objective Multivariate (BOMV) pairs also satisfy the same objectives, while being multivariate

instead. On the other hand, Many Objective Univariate (MOUV) aggregate 1 stock to 1 stock, this time satisfying not two but four different objectives. Finally, Many Objective Multivariate (MOMV) pairs are multivariate ones that are also clustered according to multiple objectives.

Fig. 3.2 illustrates the several cases.

	Number of objective functions		
Stocks per component	Single-objective (**SO**)	Bi-objective (**BO**)	Many-objective (**MO**)
Univariate (**UV**)	SO,UV	BO,UV	MO,UV
Multivariate (**MV**)	———	BO,MV	MO,MV

Figure 3.2: Case studies for different number of stocks and objective functions.

As it will be described in section 3.3.1, the objective functions to be minimized are:

- **SO**: cointegration
- **BO**: cointegration, spread's volatility
- **MO**: NZC, cointegration, spread's volatility, half-life

Serving as benchmark, Single Objective Univariate (SOUV) is the type of pair most commonly found in the literature, composed of two cointegrated stocks. This type of pair is studied in [1, 7], among others.

It is made a distinction between pairs formed by minimizing two goals and many objective functions to replicate and further examine the work in [19], one of the few experiments found in the literature that explores multi-objective optimization in the context of a PT strategy. Similarly to [19], pairs are formed with regard to cointegration and the volatility of the spread. Additionally, this thesis implements a PF framework that also explores the half-life and the NZC.

3.2 Proposed Framework - NSGA II and NSGA III

Given the exponentially growing solution space induced by multivariate pairs and the conflicting objectives that will guide the formation of pairs, a GA will be employed in the Pair Formation phase.

In fact, GAs are able to deal with a large solution space, and present a diverse solution set. In particular, an elitist GA will be applied, NSGA II and III. Rudolph [29] demonstrates that GAs converge to the global optimal solution of some functions in the presence of elitism. Elitist GAs also alleviate the inconveniences of existing Multi-Objective Evolutionary Algorithms (MOEA) - including their computational complexity or their non-elitism approach.

Some notable applications of NSGA II to portfolio management are found in [30, 31]. Moreover, [32] also uses NSGA III to select profitable stocks.

To explore whether it is advantageous to form pairs as a MOOP, NSGA II will form pairs concerning 2 objectives, and NSGA III regarding 4 objectives.

The adopted architecture of the PF module is delineated in fig. 3.3.

A set of stocks belonging to the same sector is introduced in the pairing algorithms - the traditional iterative approach, NSGA II and NSGA III - that will form pairs. The varying types of pairs are described in section 3.1.3.

Next, the newly created pairs are traded in a threshold-based model, the most common approach to the TS, that will allow to compare the quality of the pairs in a simplistic trading environment. Finally, by analyzing the pairs' performance, the answers to RSI will be drawn.

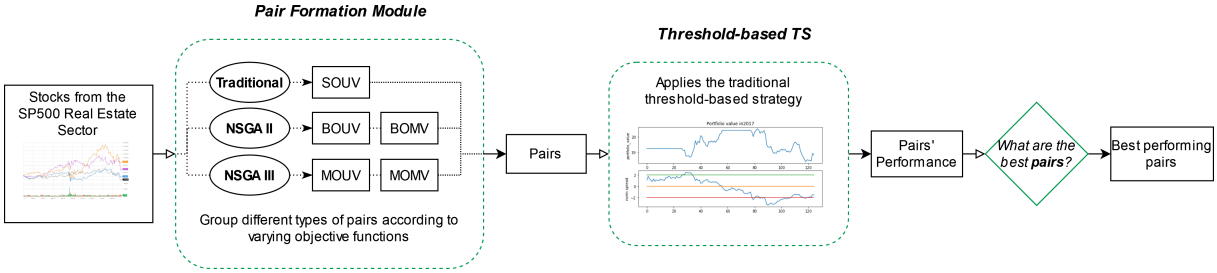


Figure 3.3: Pair Formation module architecture.

The following sections describe how the PF will be adapted to a GA, define the objective functions that quantify the quality of pairs and, at last, explore the algorithms chosen, NSGA II and III.

3.3 Problem Formulation

The multivariate PF stage can be defined as: given a set of N stocks all belonging to the same sector, select some to the first component of the pair, c^1, and other to the second component, c^2, such that the resulting pair presents considerable prospective profit.

Consider M objectives that define the quality of the pair and two binary vectors of size N, c^1 and c^2. If c_i^1 is equal to one, that means that stock i belongs to c^1. The problem may be formulated as:

$$\begin{aligned}
\min_{c^1,c^2} \quad & f = [f_1, f_2, \ldots, f_M] \\
\text{subject to} \quad & l \leq c^{1,2} \cdot \mathbf{1} \leq u && (3.2) \\
& c_i^1 + c_i^2 \leq 1, \qquad i = 1, \ldots, N && (3.3) \\
& coint(c^1, c^2) \leq 0.05 && (3.4) \\
\text{and} \quad & c_j^{1,2} \in \{0, 1\}, \quad j = 1, \ldots, N && (3.5)
\end{aligned}$$

The various objective functions, f_1 up to f_M, intend to ensure that the pair presents simultaneously profit potential and low risk.

Constraint 3.2 controls the cardinality of each component of the pair. The components must have at least l stocks, but no more than an upper bound u. This requirement is represented by the dot product of the component vectors, c^i, by a vector of ones, **1**. For a univariate pair, where each component is composed of 1 stock, both u, l would be set equal to 1. For a multivariate case, the upper limit would be greater than 1. The second constraint, 3.3, states that stock j cannot belong to both components at the same time. Furthermore, the par must be cointegrated, hence constraint 3.4. Finally, the component vectors will be composed of binary variables.

- **GA formulation**

In a GA context, each individual will represent a potential pair. As each pair has 2 components, the chromosome will have size $2N$. The first N genes concern the first component and the latter N the second. As stated previously, each gene is a binary variable. In other words, the chromosome could be interpreted as a Boolean vector: if for a given position i the gene is active, it means that stock i was added to the pair.

Figure 3.4: Example chromosome for a universe of three stocks.

Fig. 3.4 illustrates a chromosome used to form a pair from a universe of three stocks, [A, B, C], hence $N = 3$. In the configuration presented, the pair selected is (B, A).

Concerning the phenotype, the performance of a individual for each of the M objective functions will influence its survival and fitness.

Additionally, as the chromosome is composed of binary genes, the Sampling, Mutation and Crossover mechanisms will be adapted to binary variables.

3.3.1 Objective functions

But how to quantify the potential/quality of a pair and what does ”minimize risk / maximize profit” translate to? The ideal pair will be examined next.

In the one hand, the ideal pair shall offer as many and as profitable trading opportunities as possible. In other words, the pair should converge/diverge often (i). Additionally, the amplitude of the divergence must be as big as possible, to maximize the profit (ii).

In the other hand, to minimize the risk of an unprofitable trade or a rogue/divergent pair, the pair must be highly cointegrated (iii) and exhibit strong mean reversion (iv). Also, a given divergence must be met with a quick and immediate convergence (v).

The following objective functions will be used to meet the goals previously enumerated:

- **Number of Zero Crossings**

The NZC is the number of times the spread crossed its mean during the formation period. Naturally, the NZC can be indicative of strong mean reversion (iv), as implied in [7, 19], but it can also be interpreted as a measure of how frequently the pair diverges. A pair that has a high NZC displays a high number of potential trading opportunities (i).

- **Spread's historic volatility**

In finance, the historic volatility of a series is reflected by its standard deviation [33]. A spread that has a high standard deviation offers valuable trading opportunities (ii).

- **Cointegration - Engle-Granger test**

Following the majority of the literature, a measure of cointegration will be used to guarantee that the stocks grouped establish a sound and long-term proximity relationship (iii). In particular, the Engle-Granger test will be employed. This statistical method poses the null hypothesis that no cointegration exists. If the $pvalue$ of the test is close to zero, said hypothesis is refuted and the pair is considered cointegrated.

- **Half-life**

To ensure that the pair converges quickly, the half-life of the spread must be as short as possible. Minimizing the half-life not only protects against "bad behaved pairs", pairs that diverge indefinitely, but also ensures that the duration of the trade is as short as possible, minimizing the costs associated with the shorting operation. As consequence, this objective function not only concerns (v) but also maximizes profit.

3.4 NSGA II

Having described how the problem will be adapted to a GA framework, the following section focuses on the algorithm to be employed, NSGA II.

3.4.1 Overview

Since its introduction in [34], NSGA II is the reference when it comes to multi-objective optimization [35]. NSGA II is an evolutionary algorithm that avoids the pitfalls associated with the classical gradient-based optimization techniques - such as converging to a local minimum.

NSGA II most relevant and innovative features are:

- Elitist principle, i.e. the elites of a population are given the opportunity to be carried to the next generation.
- Uses an explicit diversity-preserving mechanism (Crowding distance).
- Emphasizes the non-dominated solutions.

The elitism approach leads to better and faster performance and ensures that no good solutions are "lost" along the algorithm's run. The diversity-preserving mechanism guarantees a well spread set of solutions along the optimal Pareto curve.

3.4.2 Main loop

Fig. 3.5 illustrates the main loop of the algorithm.

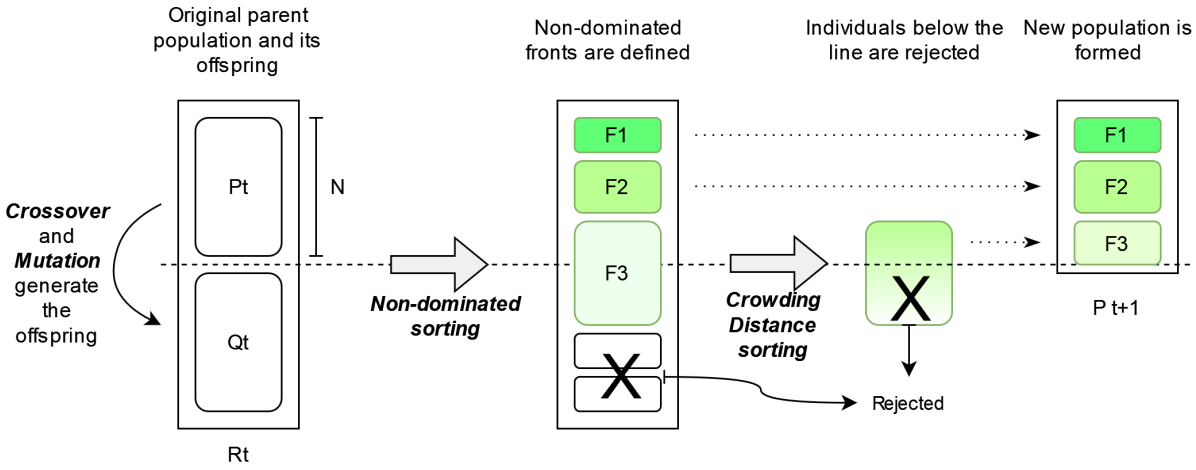


Figure 3.5: NSGA II algorithm.

Consider a parent population P in iteration t, P_t, of size N. Out of the existing P_t, an offspring Q_t is created, by mixing the genes of the best performing individuals. The entire population, R_t, is twice the size of the original population.

The entire population is then sorted, using a mechanism labeled "Non-dominated sorting" , and non-dominated fronts are defined. This technique organizes the individuals in the population by quality. The individuals in front 1, F1, have higher quality than the ones in F2, and so on and so forth. This step is represented in the middle section of the picture.

Now that the fronts are sorted, individuals are selected to the surviving population, P_{t+1}. Clearly, the higher quality individuals shall have preference, which means adding the individuals in F1 first, followed by the ones in F2, etc., until the size N is met. However, P_{t+1} shall have size N, and the entire population has size $2N$. Additionally, it is possible to observe in fig. 3.5 that the entirety of F3 does not fit in the

surviving population P_{t+1}. Consequently, only part of F3 is accepted. Using the "crowding distance" measure, the individuals in F3 are further sorted. "Crowding distance" relates to the density of solutions around the individual. Lesser dense areas are preferred, as a way to enhance diversity and ensure well spread solutions.

Finally, after having sorted the population, the individuals that exceed the desired size of the population are discarded. In the example, these less fortunate solutions are the ones below the horizontal dashed line, including the bottom part of F3. The surviving population will then transit to the next iteration where, through crossover and mutation processes, the new offspring will be generated.

3.4.3 Non-Dominated Sorting

In a multiple objective problem, *domination* can be formally defined as:

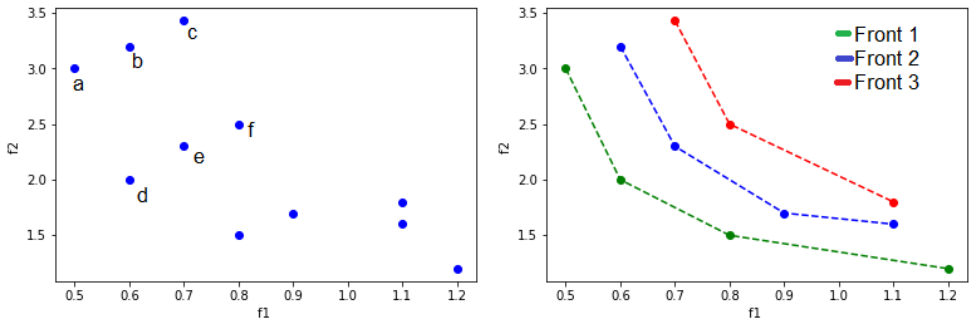
Definition 3.4.1. A(f_1^A, f_2^A) $\preceq$ B(f_1^B, f_2^B) if:

$$(f_1^A \leq f_1^B \text{ and } f_2^A \leq f_2^B) \text{ and } (f_1^A < f_1^B \text{ or } f_2^A < f_2^B). \tag{3.6}$$

In other words, A dominates B if both conditions are true:

- A is no worse than B for all objectives.
- A is strictly better than B in at least one objective.

Two entities are calculated for each individual, p: 1) domination count, n_p and 2) the set of individuals that it dominates, S_p. For a given individual, domination count refers to the number of other solutions that dominate it. Examining fig. 3.6, it is possible to observe that individual **a** dominates **b** and **c**, but is not dominated by any other solution. Consequently, its domination count would be $n_a = 0$ and the set of individuals that it dominates $S_a = [b, c]$.



(a) **(b)**

Figure 3.6: Example solutions (a) and corresponding non-dominated fronts (b).

The individuals belonging to the first non-dominated front are easy to find - their domination count is, naturally, 0. In this example, both **a** and **d** are in the first non-dominated front.

Then, for each solution p from front 1, each member of its set S_p is visited and its domination count is reduced by one. In doing so, if the domination count of any member becomes zero, it is put in a separate list. These individuals belong to the second non-dominated front. Table 3.1 illustrates the domination count and the set S_p for the most relevant individuals from fig. 3.6.

Table 3.1: Domination count and dominated set for some example individuals.

Individual	Domination count	Set S_p
a	0	[b, c]
b	2	[c]
d	0	[b, c, e, f]

Since both **a** and **d** dominate **b**, when visiting the sets S_a and S_d, n_b will be reduced by one two times. As consequence, n_b is now zero, indicating that **b** belongs to front 2.

This process is repeated until all solutions belong to a front.

Fig. 3.6 b) displays the result of non-dominated sorting for some example points.

It is relevant to note that elitism is introduced by comparing the offspring with their ”parents” in the sorting process.

3.4.4 Crowding Distance

The main diversity-preserving mechanism in NSGA II is the ”crowded-comparison” approach. This technique is employed in the population reduction step as well as during the generation of offspring. It contributes towards a uniformly spread-out Pareto optimal front by promoting individuals in less dense areas of the solution space.

The crowding distance is the Manhattan distance between the closest neighbors of a given individual. In fig. 3.7 it is possible to observe that solution A has a higher crowding distance than solution B and that indeed A habits in an area of sparse population.

The Crowded-comparison operator ($\prec_n$) evaluates the non-domination rank and the crowding distance of two solutions to encourage heterogeneity in the solution set.

Definition 3.4.2. For a given i, j with non-domination ranks i_{rank}, j_{rank} and crowding distance i_{dist}, j_{dist} $i \prec_n j$ if:

$$i_{rank} < j_{rank}, \text{or } (\ i_{rank} = j_{rank} \text{ and } i_{dist} > j_{dist}\ ). \tag{3.7}$$

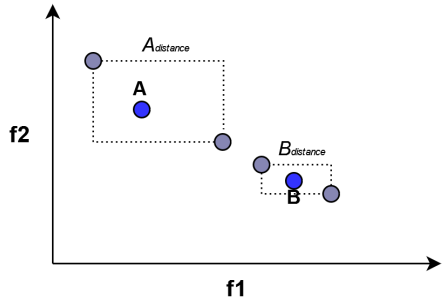


Figure 3.7: Crowding Distance measure.

3.4.5 Offspring Generation

NSGA II uses three mechanisms to generate offspring: Tournament Selection, Crossover and Mutation. The Tournament Selection is the most relevant feature, since it employs the Crowded-comparison operator to promote diversity and elitism. The Crossover and Mutation steps are no different from a generic GA. These phases will be described next.

- Tournament Selection

  The Tournament Selection procedure is rather simple but crucially important to guarantee the best and most diverse solutions. The process can be described in the following steps:

  1. Pick 2 random individuals and compare their Crowded-comparison operator
  2. Select the one with the best Crowded-comparison operator. This solution will get to reproduce and pass its genes to the following generation
  3. Repeat steps 1. and 2. to select another ”parent” (every ”child” has 2 ”parents”).

  The Tournament Selection improves performance greatly, as it implies that better solutions get to reproduce more often (since they win the tournament frequently).

- Crossover

  Having selected the two parents, the crossover mechanism is the combination of the parents' genome to generate the children's chromosome. At each coordinate of the child vector, the typical crossover function randomly selects a gene, at the same coordinate from one of the two parents and assigns it to the child.

- Mutation

  Mutation is used in GAs in order to push hypotheses toward optimal solutions and escape local minima. Generally used sparingly, the mutation method randomly changes the genes in the children's chromosome.

3.5 NSGA III - Many-objectives and the ever-growing solution space

With the increase of the number of objective functions, the number of non-dominated solutions increases exponentially. This leads to several negative consequences [36]:

- New solutions have less representation in the population as the algorithm iterates
- Diversity preservation becomes increasingly more complex
- Representation and visualization of the trade-off surface and Pareto optimal fronts are difficult.

To tackle these issues, the authors [37] develop a novel diversity maintenance technique, using reference points instead of the formerly used Crowding distance. NSGA III is an evolution of the previous algorithm that is better suited to tackle many-objective optimization problems($M \geq 3$). As result, NSGA III differs from its precursor on how to filter/discard the population and on the rules of the tournament selection.

3.5.1 Reference Points - Das and Dennis's vs Energy-based method

Deb et al. [37] state that it is too much to expect that a population-based algorithm not only converges to the Pareto-optimal front but simultaneously presents a well distributed solution set. The authors supply a set of reference points so that solutions spread across the entire solution space, i.e., so that diversified types of solutions are found.

The authors use Das and Dennis's systematic approach, that places well-spaced points in a M-dimensional simplex, that complies with:

$$z \in [0,1]^M \text{ such that } \sum_{i=1}^{\mathrm{M}} z_i = 1. \tag{3.8}$$

For instance, considering a problem with three objectives, ($M = 3$), fig. 3.8 illustrates the corresponding simplex and reference points. In the figure, the dots are the generated reference points, the gray surface is the region that satisfies eq. 3.8, and the ideal point is, naturally, the origin.

The number of points in the simplex is determined by p, which defines the number of gaps between two consecutive points along an objective axis. In fig.3.8 it is possible to observe 4 gaps along the blue line between the f_1, and f_2 axis. The number of reference points is calculated with the expression:

$$n = \binom{M+p-1}{p} \tag{3.9}$$

Revisiting the previous example problem, and selecting 4 gaps between adjacent points on an objective axis, ($p = 4$), eq. 3.9 yields 15 reference points. Observing fig. 3.8, it is possible to verify that there are indeed 15 reference points.

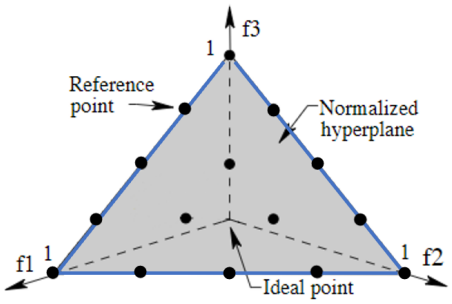


Figure 3.8: Das and Dennis's reference points. Adapted from [37].

However, in later works, the authors in [38] point some drawbacks of the Das and Dennis's method:

- As the number of gaps, p, increases, the number of reference points raises exponentially.
- Most reference points will lie on the frontier, the line between objective axis (highlighted in blue in fig. 3.8).

To ensure a well spread set of reference points, it is desirable that each gap is as small as possible, meaning selecting a high value for p. In addition, the authors state that the size of the population must be somewhat equal to the number of reference points. As consequence, the population would also grow exponentially, which increases the computational power and time needed massively.

The second disadvantage is evidenced in fig.3.8: of the 15 total points, only 3 are not on the blue lines. This factor leads to misrepresentation of the center reference points. Their proportion to the outer frontier points is rather small, placing little to no importance in the center directions.

The authors propose instead an innovative physics-inspired energy-based approach [38]. Multi-body and interacting physical systems eventually settle on a state with the minimum overall potential energy. Given that potential energy is inversely proportional to the distance between bodies, the method places the points that minimize the Riez s-dimensional Energy:

$$U_T(\mathbf{z}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{\|\mathbf{z}^i - \mathbf{z}^j\|^s}, \mathbf{z} \in \mathbb{R}^{n\times M}, \tag{3.10}$$

where $\mathbf{z}^{i,j}$ are M-dimensional points and $\mathbf{z}$ the entire matrix that encompasses all the n points. It should be noted that the authors state that it is not clear how the dimension s should depend on the number of objectives, (M). Nonetheless, they find by trial and error that $s = M^2$ produces the best results.

Summarizing, the energy-based method assigns the points in such manner that the distance between all points is as big as possible. This method is superior to Das and Dennis's one in the sense that it allows the user to set any given number of reference points, instead of having to worry about the rising number

of reference points and the consequential immense computational power needed. For such reasons, the energy-based method will be used in this work.

3.5.2 Diversity-Preserving Operation

Following the description of how to obtain reference points, it is now needed a mechanism that associates said references to the population. As result, the proposed NSGA III, in addition to emphasizing non-dominated solutions, also prioritizes population members which are in some sense associated with each of these reference points. This procedure will act when sorting and filtering the population, and when selecting individuals to reproduce (in the Tournament Selection).

When associating individuals with normalized reference points, it is necessary to also normalize the individuals. As so, the solutions are normalized at each iteration. This has the added benefit of allowing NSGA III to solve problems with differently scaled objective functions.

Subsequently, the solution points are associated with the reference directions - a given individual is related to the closest reference. Consider the example of having a front that does not fit in the surviving population, P_{t+1}, where the niche preservation mechanism acts as follows:

1. Select the reference direction with fewer surviving individuals. This is the least represented direction.
2. Add the closest solution related to that reference to P_{t+1}.
3. Repeat the previous steps while P_{t+1} is not complete.

Diversity is preserved and accentuated since the least represented direction is addressed first.

3.6 Conclusion

In this chapter, a new PF technique that groups stocks as a MOOP was presented. The problem was described and then adapted to a GA formulation. In particular, NSGA II and III will be employed to create pairs. Next, the different types of pairs that reflect the research questions in RS I were enumerated. Pairs will vary in the number of stocks in its composition and will be formed according to different objective functions. Finally, the NSGA II and III algorithms were described in detail.

Table 3.2 summarizes some main characteristics of the adopted PF framework.

Table 3.2: Key Points from the PF framework.

Pairs Characteristics	Motivation
Multivariate Pairs	Increasing the number of assets in hand reduces volatility;
Created with MOOP	Maximize profit and minimize risk;
Objective functions	Reasoning
Min. cointegration	The most used condition to form pairs; Guarantees a long-term relationship;
Max. spread's volatility	Wide movements in the spread increase profit;
Max. NZC	Ensures strong mean-reversion, minimizes risk;
Min. Half-life	A quick convergence implies more trading opportunities;
PF Algorithms	Key Points and application
GAs	Able to deal with massive solution spaces; Do not get trapped in local minima;
NSGA II	The reference when it comes to bi-objective optimization; Used to create BOUV and BOMV pairs;
NSGA III	The successor of NSGA II, better equipped to deal with many-objectives; Used to create MOUV and MOMV pairs.

4

Proposed Trading Module

Contents

4.1 Forecasting-Based Trading Stage 39

4.2 Forecasting TS architecture 40

4.3 ARIMA 41

4.4 XGBoost - Extreme Gradient Boosting 42

4.5 Conclusion 46

The present chapter proposes a new trading framework based on ML, in order to improve the trading stage of a PT strategy. This Forecasting-based Trading Stage will answer the research question:

- Can a **ML-based trading strategy** improve PT?

4.1 Forecasting-Based Trading Stage

The traditional threshold-based trading model attributes the entirety of its success, (or lack of), to the thresholds that define the entry and exit points of the strategy. Surpassing the hardship of finding/guessing the thresholds that maximize the overall profit would be a considerable improvement. Furthermore, the traditional strategy only trades the signal in the direction of convergence, towards the mean. Consequently, the portfolio value keeps decreasing while the pair is still diverging, and only increases when convergence is reached. Additionally, this means that there is a considerable amount of time when the spread is not being traded (while it has not activated one of the thresholds), which implies that, if the signal could be traded in all directions, profits would increase.

Stock price forecasting is an important and widely researched topic in financial science [39]. According to the Efficient Market Hypothesis, the current price of an asset reflects all the available and most recent information - ”the market is always right”. As such, by studying past values of stock prices, and implicitly examining the history and behavior of the stock, some insight could be gained into the future.

As previously mentioned, a PT strategy would benefit from anticipating the spread's movements, to select the best possible moments to enter and exit the market. To address such problem, the model developed will predict the day-ahead value of a pair's spread, and this prediction will define the market entry points.

The forecasted spread will be used to detect trends in the pair's movement. While the predicted value for tomorrow is superior to today's real observation, it is considered that there is an uptrend, and the strategy will long the spread. Likewise, while it is predicted that the spread will decrease, i.e., while the day-ahead value is smaller than the real current one, the spread presents a downtrend, and the strategy will short the spread.

Fig. 4.1 illustrates the strategy described, assuming that the forecast has perfect accuracy.

Formally, the strategy tracks the predicted change in the spread's value, defined as

$$\Delta_{t+1} = \hat{s}_{t+1} - s_t, \tag{4.1}$$

and recommends a Long/Short position following:

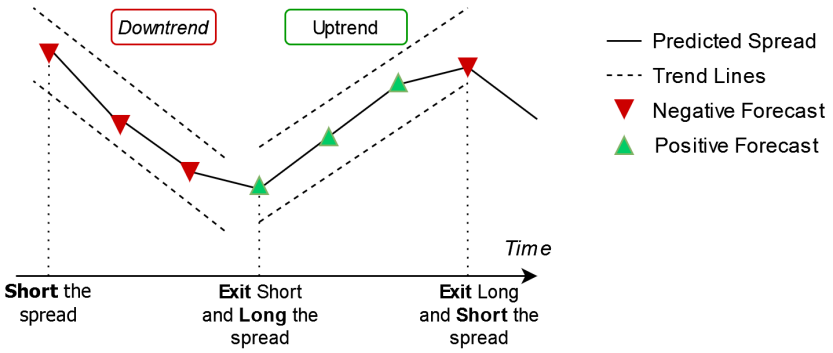


Figure 4.1: Trend-based forecasting strategy.

$$\text{Recommended Position} = \begin{cases} \text{if } \Delta_{t+1} > \text{Transaction Costs}, \text{Long position}, \\ \text{if } \Delta_{t+1} < -\text{Transaction Costs}, \text{Short position}, \\ \text{otherwise, remain outside the market.} \end{cases}$$

The strategy then monitors the recommended position for each day and acts when there is a change in the type of position, meaning that there was an inversion in the direction of the trend. Revisiting fig. 4.1, where the trader's actions are displayed in the bottom part of the figure, when there is a transition from a Short to a Long position, the trader is called to act and change its position. In this case, it would go from shorting the spread to buying the spread.

The spread forecasting will be implemented as a regression problem, since it must be ensured that the predicted Δ_{t+1} is bigger than the transaction costs associated with the operation. This method is considered more robust than forecasting the spread as a classification problem, where the spread is predicted to either grow or decrease.

4.2 Forecasting TS architecture

The structural design of the proposed TS is presented in fig.4.2. Firstly, the forecasting models, ARIMA and XGBoost, are fitted to the spreads to forecast. Then, the day-ahead prediction is generated. Next, the real and the predicted spreads are analyzed by the Forecasting-based strategy, that monitors potential inversions in the spread's movement, the trend. Finally, the performance of the trading system developed will allow to test if a ML-based approach contributes positively to the trading stage.

The following sections describe the techniques used to forecast the spread.

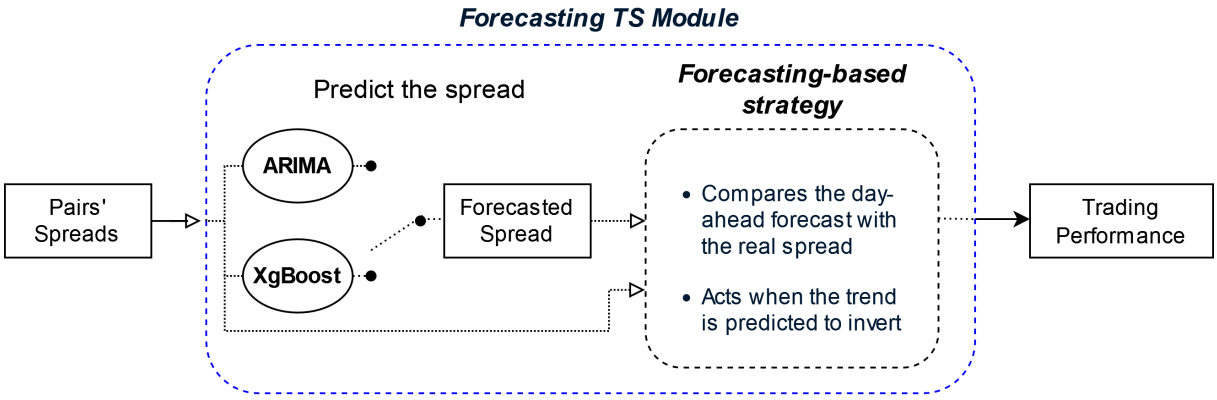


Figure 4.2: Forecasting TS module architecture.

4.3 ARIMA

The ARIMA model relies on statistical properties, and is characterized by its theoretical simplicity, its ability to capture trends and, finally, the fact that it is straightforward and fast to train. This model was selected due to its transparency and proven robustness in a wide field of forecasting applications [40–42]. Despite not being able to capture non-linear patterns in data, it will provide an interesting comparison to the other ML model selected, questioning whether the additional complexity is noticeable and justifiable in stock market forecasting.

In the following sections, the ARIMA model will be decomposed in its building blocks and further analyzed.

4.3.1 AR – Auto-Regression

The autoregression component of ARIMA describes the current value, y_t, as the sum of a constant, c, plus the p lagged/past values multiplied by their coefficient α_n and a white noise term, ϵ_t.

$$y_t = c + \sum_{n=1}^{p} \alpha_n t_{t-n} + \epsilon_t. \tag{4.2}$$

4.3.2 MA – Moving Average

The second major component, the moving average, should not be confused with a traditional rolling average. It is instead the past n observations of white noise. It can be described as in (4.3), where θ_n is the coefficient associated with error term ϵ_{t-n}.

$$y_t = c + \sum_{n=1}^{q} \theta_n \epsilon_{t-n} + \epsilon_t. \tag{4.3}$$

4.3.3 ARIMA - Adding Integration

ARMA models are applicable to stationary series only, that is, series with mean and variance that do not change with time. Ideally, the spreads to be predicted are stationary, since that is the core concept of PT: find a pair composed of two stocks with a linear combination that results in a stationary spread, meaning that the pair is cointegrated. However, although spreads are expected to be stationary in the formation period, there are no guarantees that this condition will still be present in the trading period, as the future is not determined by the past. To address this problem, ARIMA models introduce a differentiation step to the data, describing the change of y_t instead of the original y_t. In other words, an ARIMA model is an ARMA model on the differentiated time series. Finally, the complete ARIMA model is described by

$$d_t = \frac{d}{dt} y_t = c + \sum_{n=1}^{p} \alpha_n d_{t-n} + \sum_{n=1}^{q} \theta_n \epsilon_{t-n} + \epsilon_t. \tag{4.4}$$

Three parameters must be defined by the user:

- **p**: the number of past observations to analyze;
- **d**: the order of integration of the signal to represent, or the number of times the signal must be differentiated so that a stationary one is obtained;
- **q**: the number of lagged values used in the moving average. This parameter is commonly set equal to zero, meaning that only the current error term is considered.

4.4 XGBoost - Extreme Gradient Boosting

Introduced in 2014, XGBoost [43] provides a novel take on gradient boosting frameworks. Characterized by being highly flexible and capable of dealing with massive and complex data sets, XGBoost will be used to forecast the spread, based on the findings in [44], where it is concluded that frameworks that rely on trees produce superior results compared to ANNs such as RNNs. Additionally, tree-based architectures are faster to train.

4.4.1 XGBoost Overview

In a nutshell, XGBoost successively creates a collection of trees, each one of them trying to improve where the previous ones failed. The contribution of all trees is then used to make a prediction. If such prediction is deemed not accurate enough, more trees are formed.

As in gradient boosting, XGBoost gives a prediction model in the form of an ensemble of weak learners. However, it uses an unique type of regression trees, commonly named XGBoost trees. Further-

more, it works as a Newton-Raphson method in the objective function space, instead of implementing a gradient descent-based method as the gradient boosting algorithm.

Similarly to common decision trees, the algorithm employs a scoring function to assess the quality of the tree structure. Commonly known as the Similarity or Structure Score (SS), it can be compared to the Gini impurity metric in decision trees. The Structure score for leaf j is given by:

$$\text{Structure Score} = \frac{(\text{Sum of residuals in leaf } j)^2}{\text{Number of residuals in leaf } j + \lambda}. \tag{4.5}$$

An overview of the algorithm is presented next.

1. Start with an initial prediction and calculate the error - the residuals - to the real observations.
2. Build a tree to describe the residuals, based on the data features.
   (a) Group all the residuals in the root node.
   (b) Split the residuals into new leaves, creating branches, while the SS is improved or the max depth is not reached.
3. Prune the newly constructed tree.
4. Compute the output value for each leaf.
5. Make a new prediction, compute the residuals and repeat steps 2 to 5, until no improvement is made or a maximum number of iterations is exceeded.

4.4.2 Formal definition

Consider a data set with N elements, each element with x features. The XGBoost algorithm will form a collection of K trees, f_1 up to f_K. Each tree assigns a leaf value, w to the features given. In other words, $f(x) = w$. Naturally, the trees' structure varies, each tree yields different output values to the same features.

The model generates a prediction considering the contribution of all K trees,

$$\hat{y} = \sum_{k=1}^{K} f_k(x). \tag{4.6}$$

The model is trained in addictive fashion, and every new tree will try to predict the residuals of the previous trees. Effectively, each new tree tries to improve where the others have failed.

Let $\hat{y}_i^{(t)}$ be the prediction of the i-th instance at time t. The new tree shall be constructed in such a way that the overall objective function is minimized. The objective function is the distance between

all predictions and real observations. As such, the new tree, f_t, will be formed to minimize the overall regularized loss

$$L^{(t)} = \sum_{i=1}^{N} l(y_i, \hat{y_i}^{(t-1)} + f_t(x_i)) + \Omega(f_t), \tag{4.7a}$$

$$\text{where } \Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||^2. \tag{4.7b}$$

The function $\Omega(f)$ affects all the T leaves of the tree f_t, penalizing the formation of too many leaves with parameter γ, and applies a Ridge Regression regularization over the leaf outputs w. The regularization mechanisms of XGBoost will be further explored in the following sections.

The loss L is approximated by its second order Taylor series to easily optimize the objective, giving raise to the comparisons with the Newton-Raphson method:

$$L^{(t)} \simeq \sum_{i=1}^{N} [l(y_i, \hat{y}^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t), \tag{4.8}$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ are the first and second order gradients of the loss function, following the notation in the original manuscript.

Eq. (4.8) can be rewritten as follows, by defining I_j as the set of residuals in leaf j, removing the constant term $l(y_i, \hat{y}^{(t-1)})$ (does not interact with the new tree f_t), and remembering that $f(x) = w$:

$$\tilde{L}^{(t)} \simeq \sum_{j=1}^{T} [(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T. \tag{4.9}$$

It is now possible to compute the optimal value for each leaf j, w^*, by differentiating the loss (4.9) and setting it equal to zero:

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}, \tag{4.10}$$

and the optimum minimum value for $\tilde{L}$ is obtained with:

$$\tilde{L}(t) = -\frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{i \in I_j})^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T. \tag{4.11}$$

Since the loss function employed by the authors is the traditional $l(y_i, \hat{y_i}) = \frac{1}{2}(y_i - \hat{y_i})^2$, its first order gradient will be $g = -(y_i - \hat{y_i})$, the error, or the residual between the real observation and the prediction

multiplied by -1. The second order derivative will be $h = 1$. This allows to simplify eq. (4.10):

$$w_j^* = \frac{\sum_{i \in I_j}(y_i - \hat{y_i})}{\sum_{i \in I_j} 1 + \lambda} = \frac{\text{Sum of residuals in leaf } j}{\text{Number of residuals in leaf } j + \lambda}, \tag{4.12}$$

and (4.11):

$$\tilde{L}^(t) = -\frac{1}{2}\sum_{j=1}^{T} \frac{(\text{Sum of residuals in leaf } j)^2}{\text{Number of residuals in leaf } j + \lambda}. \tag{4.13}$$

Paraphrasing eq. (4.13), the optimal objective function value is obtained by forming a tree that maximizes the Structure Score for all its leaves. This justifies the use of the Structure Score when creating branches (step 2. (a) of section 4.4.1).

Additionally, the output value of each tree used in step 4 in section 4.4.1 is computed with eq. (4.12). When no regularization is used, $\lambda = 0$, the output value of a leaf is simply the average of the residuals associated with said leaf.

4.4.3 Regularization

Tree-based algorithms such as Decision Trees, are notorious for suffering over-fitting. In order to reduce this phenomenon, XGBoost employs some regularization mechanisms that are described next.

- λ

The user defined parameter λ is intended to reduce the prediction's sensitivity to individual observations. For values of λ greater than zero, the Structure Score and the output value of the leaves are reduced. Additionally, the amount of decrease is inversely proportional to the number of residuals in the node, i.e., the more residuals are in the node, the less affected the node is. This parameter penalizes leaves that describe the fewer residuals, as it is desirable that leaves represent a large range of residuals as to be flexible and adaptable when faced with unforeseen data.

- γ

Pruning techniques ensure that trees tend to generalize better on new data. Parameter γ is also set by the user and influences the pruning mechanism. As seen in eq. (4.11), the number of leaves T in a newly formed tree negatively influences the objective function. During the pruning of the tree, nodes whose Structure Score is not bigger than γ are removed, since a tree with more leaf nodes fits very well to the training data, thereby resulting in over-fitting. As such, gamma promotes trees with the fewer leaves possible.

- Shrinkage

Analogous to a learning rate technique, shrinkage, η, reduces the weight of each individual tree in a prediction, $\hat{y}^{(t)} = \hat{y}^{(t-1)} + \eta f_t(x)$, leaving room for improvement for future trees.

- Early Stopping

During the training process of ML models, the error to the training data decreases steadily as the model adapts itself to better describe the training data. However, in doing so, the model loses flexibility to describe unseen data, it over-fits, and the error to the validation set consequently begins to increase. Early Stopping monitors the validation error and brings the training to a halt if the error has not improved in the past epochs, thus reducing over-fitting.

4.5 Conclusion

In this chapter, the motives that justify a more refined ML-based approach to the TS were firstly presented. Secondly, the proposed trading strategy was introduced. Finally, the forecasting models, ARIMA and XGBoost, were reviewed, along with several regularization techniques. Table 4.1 highlights the most relevant features of the TS developed.

Table 4.1: Main characteristics of the proposed Trading Model.

Strategy Characteristics	Reasoning
Forecast the spread	Anticipates trading opportunities; Improves the ”blind” and rigid traditional trading model;
Act when the trend inverts	Defines more precise entry points; Allows to trade the spread in both directions.
ML models	Reasoning
ARIMA	Used in a wide range of applications, proving its robustness; Will serve as a benchmark to more complex models;
XGBoost	Flexible and able to deal with large data-sets; Captures complex patterns in data.

5

Experiment Design

Contents

5.1 Data set - S&P 500 Real Estate sector 49
5.2 System Architecture 50
5.3 Pair Formation Stage 51
5.4 Forecasting-based Trading Stage 52
5.5 Trading Simulation 55
5.6 Evaluation Metrics 58
5.7 Conclusion 59

Having presented the framework that will be employed, both to address the PF stage and the forecasting-based TS, this chapter discusses some relevant aspects regarding the implementation. In particular, the data set, the back-testing setup and trading rules will be addressed.

5.1 Data set - S&P 500 Real Estate sector

The S&P 500 is one of the most famous and well-known market indexes. It comprises the 500 biggest publicly traded companies on American stock exchanges. Given its popularity and the fact that its data is widely available, this work will test the strategies developed using S&P 500 financial information. Additionally, as mentioned in the Literature Review, several pairs trading related works trade companies from the S&P 500, facilitating the comparison of the results obtained in this thesis with others.

Stocks belonging to the Real Estate sector will be studied. The Real Estate sector is composed of 30 companies and constitutes 2.9% of S&P 500 and, as the name suggests, includes Real Estate Investment Trusts (REITs), as well as realtors and other companies. Stocks will be selected within the same sector to ensure that a relationship between two stocks is not coincidental, but a consequence of the fact that they are influenced by the same economic factors. Additionally, restricting the set of stocks eases the problem of the exponentially growing number of possible combinations, introduced by using multivariate pairs.

5.1.1 Data partition

Following the traditional method of partitioning the data into a training/testing set, the data will be split in two periods: the formation and the trading period. The formation period represents the time interval prior to a trade. Pairs will be formed and selected, and the forecasting algorithms will be trained. The trading period simulates a real time trading environment: the strategies act upon unforeseen information as if it was a real time transaction. The present work attempts to reproduce the hardships of implementing a real time trading strategy as best as possible, as it will be further described in following sections.

To further validate the experiment, it is desirable to subject the algorithms to different market conditions. As the data studied is a time series, therefore it is sequential, a ”Walk Forward Validation” technique will be applied, instead of the typical ”k-fold cross-validation” commonly found in ML.

Fig. 5.1. illustrates the forward moving window. A 3-year training period will be used, based on the assumption that selecting pairs across a relatively long time period reveals long-lasting associations. Then, the formation techniques are tested in the next year.

The time frame selected for this work spans from 2014 to 2021. The test years are highlighted in bold in fig. 5.1, 2018 to 2021. The adopted period is considered relevant since it is recent - presenting

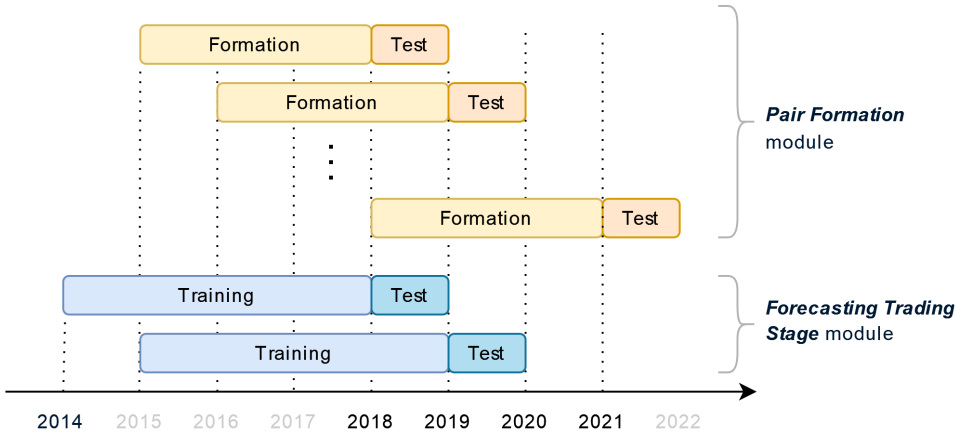


Figure 5.1: Sliding window periods.

the challenge of the volatile modern stock market - and because it includes one major economical event - the crash of March 2020 consequence of the COVID-19 pandemic.

The Forecasting-based Trading Stage applies a 4-year training period followed by a 1 year test period. This process is executed for both algorithms, ARIMA and XGBoost.

5.2 System Architecture

The entire research system's architecture is displayed in fig. 5.2, combining the PF and the Forecasting TS module.

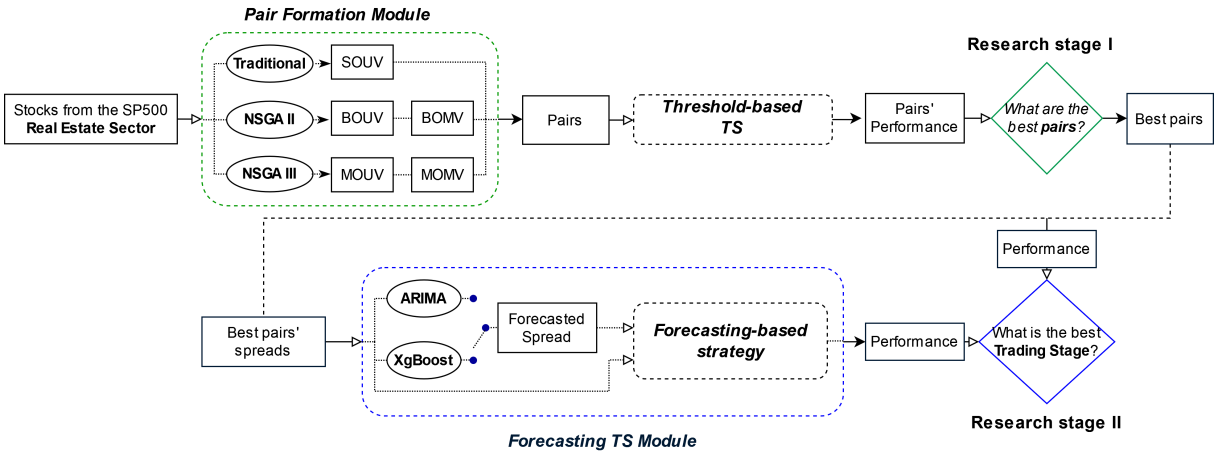


Figure 5.2: System architecture.

- **Research Stage I:**

To address RS I, different types of pairs are formed within the Real Estate sector, either by the traditional

iterative approach, NSGA II or III. The best type of pairs is determined by analyzing their performance in a simplistic threshold-based strategy.

- **Research Stage II:**

The best performing pairs from the previous stage are introduced to the Forecasting TS Module. The spread is predicted for each pair, either with the ARIMA or XGBoost models. The forecasting-based strategy then uses the predicted spread to act upon the market. Finally, to assess whether the forecasting strategy is superior to the standard threshold-based strategy, the performances are compared.

5.3 Pair Formation Stage

The following section details the hyperparameter selection of NSGA II and III, used to select pairs, and the implementation of the benchmark trading strategy, used to determine the best pairs.

5.3.1 NSGA II and NSGA III

NSGA II will be employed to form bi-objective pairs, and NSGA III will be applied to select pairs according to many-objective functions.

Both NSGA II and III were implemented using *pymoo*, a MOOP dedicated Python library. This framework was chosen since the original author of the algorithms contributed to the design of these modules [45].

A population of 50 individuals was selected, following the experiment in [19]. The algorithm iterates for 80 generations, as the evolution of the objective space stalls above this epoch. It is evident that the ideal termination criterion would be defined by monitoring the objective space and terminating the algorithm when no improvement is observed for a certain number of generations [46]. However, this method would require a much larger amount of time to converge. Appendix B further justifies this decision.

The chromosomes are composed of binary variables, that indicate whether a stock belongs to the pair or not. By modifying the sampling, crossover, and mutation mechanisms, the NSGA algorithms were adapted to a binary problem. Due to pymoo's modular nature, the aforementioned methods were easily substituted by a Binary Sampling, a Bit Flip mutation and a Two Point Crossover, all from the pymoo library, and the default probabilities were applied. Additionally, NSGA III employs an energy-based method to generate reference directions, Riesz s-Energy.

5.3.2 Threshold-based Trading Strategy

The different types of pairs stipulated in section 3.1.3 and their respective formation techniques will be evaluated experimentally, by applying a common trading strategy to all different pairs. The naive threshold-based strategy will be applied without much thought or optimization. The objective of the experiment shall be comparing the different types of pairs, not to beat the market. Table 5.1 displays the thresholds that dictate the strategy.

Table 5.1: Threshold strategy parameters.

Threshold	Value
Long	$\mu_s - 2\sigma_s$
Short	$\mu_s + 2\sigma_s$
Close	μ_s

Quite simply, when the spread reaches one of the thresholds, a position is opened and when it reverts to its mean, the position is exited, hopefully with profit. The spread's standard deviation, σ_s, and mean, μ_s, are calculated with past values, in particular with observations of the past 3 months. The period considered in the calculation of these parameters influences greatly the outcome of the strategy. Section 5.5 stresses the importance of guaranteeing that only past values are used in the calculation of σ_s, μ_s.

Another common approach found in the literature consists in normalizing the spread as a *zcore*, simplifying the thresholds to -2, +2 and 0.

5.4 Forecasting-based Trading Stage

Next, the selected hyperparameters for the forecasting models are justified. Some details of the implementation of XGBoost are pointed, including the features used and the regularization technique, early stopping. Additionally, the rationale behind forecasting the spread is exposed and experimentally supported.

5.4.1 ARIMA - hyperparameter selection

ARIMA is characterized by being a parametric model, relying on a set of hyperparameters set by the user, that reflect and explain the underlying structure of the data. As such, three parameters were defined: p, d, q. Parameter p corresponds to the number of lag observations considered, d to the order of integration of the data and, finally, q determines the size of the window employed in the calculation of

the moving average. The parameters were chosen manually, based on the analysis that follows of the signals to forecast. Fig.5.3 displays an example spread, FRT-EQR, its trend and seasonal components.

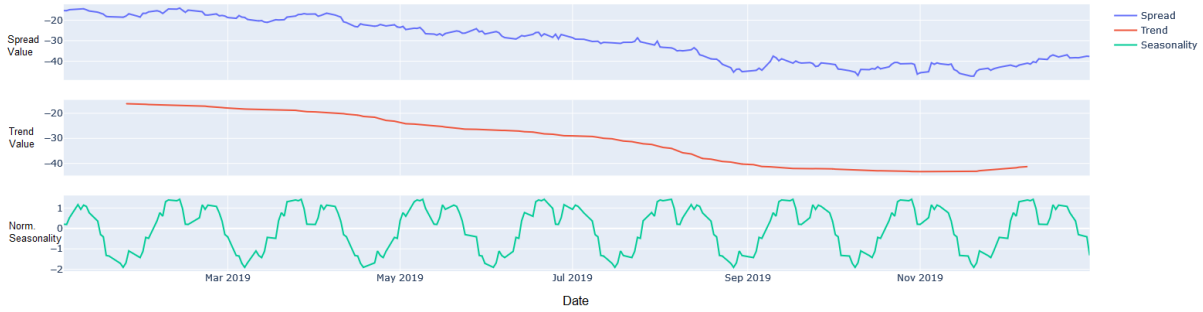


Figure 5.3: Trend and seasonal components of FRT-EQR in 2019.

Observing the trend plot in fig.5.3, a clear downtrend is observable, meaning that the spread is not stationary: its mean does not remain the same as time passes. Hence, the order of integration will be set to one, $d = 1$. It could be pointed that the spreads should be stationary, fluctuating between some well-defined thresholds, since that is the ideal condition for a PT strategy that the pairs formed should satisfy. However, test spreads rarely are stationary, as it will be exposed in the Results chapter.

Regarding parameter p, the number of past days considered in the model, there is tradeoff between large values of p, able to capture the behavior of the signal during long periods of time, and small values, that allow the model to swiftly react to changes in the stock's prices. The best results were obtained for $p = 5$, but the differences in performance were not meaningful. Additionally, the model takes much longer to converge with larger values of p.

Finally, q is set to zero, $q = 0$, so that only the current error term is considered.

To conclude, the cointegrated spread will be predicted for each pair, using an ARIMA with configuration $(p = 5, d = 1, q = 0)$ to perform the day-ahead forecast.

5.4.2 XGBoost - hyperparameter selection

The original python implementation of XGBoost [43] was used to forecast the spread as a regression problem.

Since XGBoost is capable of dealing with large and complex data sets and benefits from using many features, several features besides the stock price will be given to the model, in order to construct a more complete representation of the data. As so, to take advantage of XGBoost's flexibility and adaptability, the model will be given the following features[1]:

[1]The Moving (SMA) and Exponential (EMA) Averages, Relative Strength Index (RSI) and the Moving Average Convergence/-Divergence (MACD) are popular technical analysis indicators.

- $SMA_{5,\ 10,\ 15}$
- EMA_{9}
- RSI
- MACD

XGBoost requires the user to define some hyperparameters, namely the number of estimators or regression trees, $n\_estimators$, and its maximum depth, $maxdepth$ among others.

To determine the best parameters to forecast the spread, a grid search method was employed. The resulting best values found are: $n\_estimators = 400$, $\eta = 0.05$, $maxdepth = 8$, $gamma = 0.005$.

To avoid over-fitting, the algorithm monitors the validation error and stops the progress if no improvement is found. Fig. 5.4 shows the effect of over-fitting observed in early stages of the development, that justified the use of early stopping. The early stopping mechanism has a patience of 5 epochs.

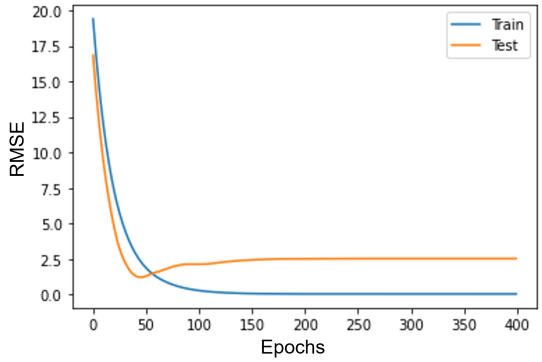


Figure 5.4: Evolution of the *rmse* metric for the spread DLR-AVB

5.4.3 Train and Test splits

The training/testing split used is $(80\% - 20\%)$. Since one entire year will be predicted, the training period encompasses 4 years.

Given the sequential dependence intrinsic to time series, a walk forward validation is used for both models. For each N days of the testing period, the model is fitted to past observations and the day-ahead prediction is computed.

It is desirable that N is as small as possible, that is, that the model is updated very frequently. Fig. 5.5 displays the superior performance yielded by fitting the XGBoost model every 5 days, in comparison with 50 days.

However, fitting the XGBoost model is considerably slower compared to the ARIMA. As consequence, there is a tradeoff between computational time and the size of N. The value $N = 5$ was chosen, since it produces the best results (no noticeable improvement is observed for smaller N), while still maintaining the computational time needed tolerable.

In summary, the model is fitted each day for the ARIMA, and each 5 days for XGBoost.

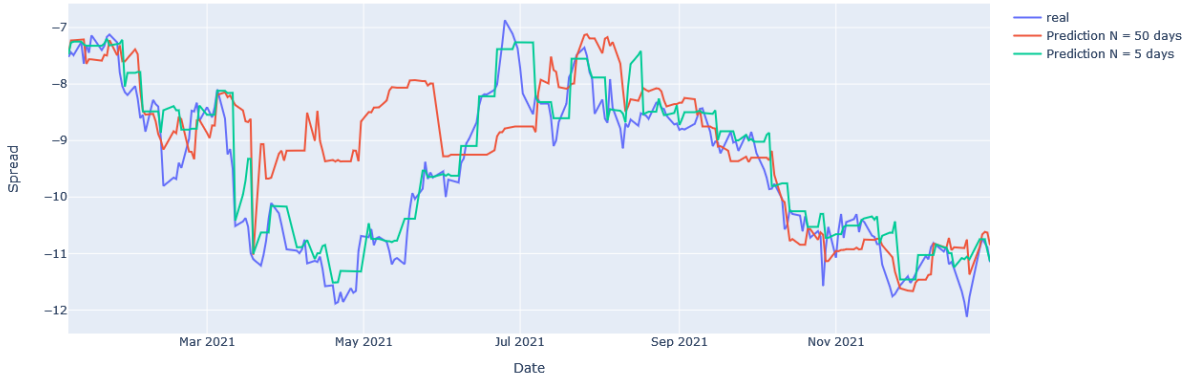


Figure 5.5: XGBoost forecasting of KIM-ESS for different N.

5.4.4 Why predict the spread and not the stocks individually?

A very pertinent question could be raised: ”why predict and trade the spread and not a single stock individually?”. Indeed, if a trader trusts their forecasting system, why not trade a single stock and reduce the complexity of the system?

The reality is that no forecasting system is fully accurate or precise. Trading two stocks, i.e. trading the spread, protects the trader against sudden movements or occasional particularly inaccurate forecasts. In fact, forecasting and trading the spread guarantees that the trader's position is always hedged: if one of the stocks moves contrary to the forecast, the other should amortize the loss.

Another relevant question should be posed: is it better to predict the spread directly or predict two stocks that constitute a pair and then compute the forecasted spread as $\hat{s} = \hat{c}_1 - \beta\hat{c}_2$?

The best results were obtained while predicting the spread directly. The reader may refer to appendix B, where a comparison table is presented. Although the performances do not differ by much, predicting each stock individually propagates the error of each forecast into the predicted spread, and consequently results in a poorer performance. Furthermore, predicting each stock individually doubles the computational effort required.

5.5 Trading Simulation

The following section highlights the implementation features that ensure some resemblance with a real trading environment, namely the transaction costs and the normalization of the spread using past values only.

5.5.1 Transaction Costs

The costs and fees associated with each operation in the market silently erase the potential profit and should not be ignored. The back-testing simulation considers shorting rental fees, market slippage and operation costs. Table 5.2 displays the costs for the mentioned types, based on the estimations in [47].

Table 5.2: Transaction costs considered.

	Explicit	Implicit	Short Rental
Description	Broker commissions, exchange fees.	Slippage, market impact, delays.	Daily fee applied to a short position.
Rate	8 bps	20 bps	1 % per annum

Explicit costs are the direct costs of trading, including broker commissions, transaction taxes, stamp duties, and exchange fees. Implicit costs are indirect, and include phenomena such as the slippage - the difference in price from the moment the trade is requested to the instant when it is truly executed [48].

5.5.2 Time series normalization and Look Ahead Bias

A crucial but frequently overlooked step of a PT strategy is the spread normalization. The test spread, s, is normalized according to:

$$Z = \frac{s - \mu_s}{\sigma_s}, \tag{5.1}$$

and it is of utmost importance that both μ_s and σ_s are calculated using only past values.

Looking into future values results in Look Ahead Bias, and it is far from a trivial detail: μ_s and σ_s determine the entry/exit thresholds for the strategy and, in a strategy where it is desirable to define the best occasions to enter and exit the market, knowing the range of values of the future signal constitutes an unfair advantage.

Ultimately, the trading results are positively (yet incorrectly) influenced: the incorrectly normalized spread will offer better and more frequent trading opportunities.

Fig. 5.6 shows the same spread normalized considering the historical mean and standard deviation and the ones considering future values. In particular, the pair ESS-DLR is normalized using the past 1 year mean and standard deviation (blue line), and using future values (red line). It is possible to observe that, while the incorrectly normalized spread completes 3 trading cycles, the spread normalized considering past values only offers 1 trading opportunity, with the entry during January and the exit at the end of October.

Normalizing the spread with d past values adds a new degree of complexity to the threshold trading strategy, as there is now the need to select the number of days considered. There is a tradeoff for the possible values of d: d too large will not be able to follow the ever-changing behavior of the spread; values too small will change the thresholds too often, overreacting to temporary changes in the spread.

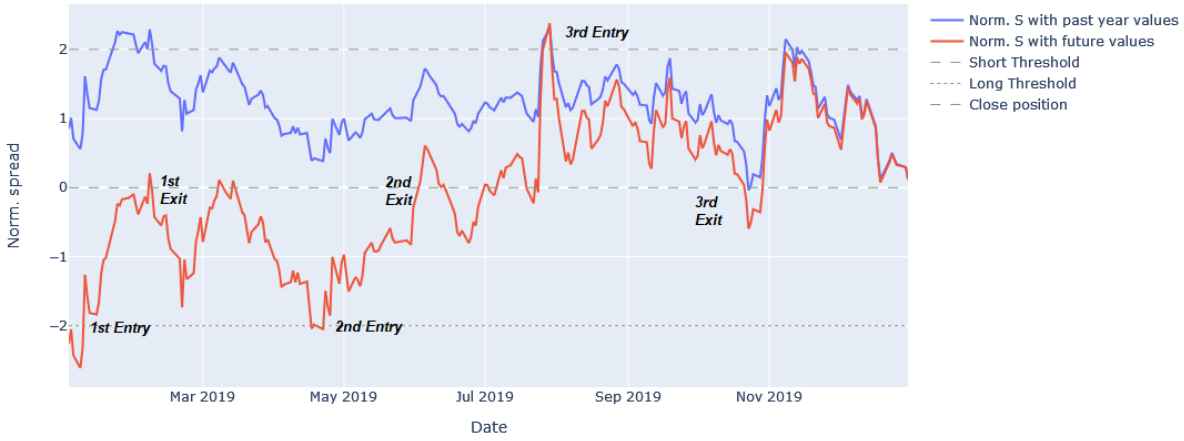


Figure 5.6: Comparison of the normalized ESS-DLR spread using past and future values.

The best value for d, $d = 63$, or approximately three trading months, was obtained by iteratively testing different normalization periods. The experiments can be observed in appendix B.

5.5.3 Trading Portfolios

Although the threshold-based and the forecast-based traders would act in different occasions - the threshold strategy acts when the normalized spread touches a threshold, and the forecasting-based when the predicted trend inverts direction - the management of the trading portfolios is the same. A portfolio is created for each one of the five types of pairs, in the PF module, and for each of the forecasting algorithms, in the Forecasting-based Trading Stage. Fig. 5.7 depicts the portfolio management during the experiment.

An initial symbolic value of 1000$ is attributed to each portfolio. Caldeira and Moura [23] propose a self-financing portfolio, that does not require the investor to finance the strategy with their own capital, since the value of the short operation covers the cost of buying the other component. Despite being enticing, this cashless strategy could not be applied in practice, since it is always required to have a guarantee - a collateral - when shorting a stock.

As the experiment progresses, the amount earned/lost during a trading year is propagated into the next year. If, for example, the first period yields a loss of 5%, the portfolio in the second year will only have 950$, instead of the original 1000$. This method simulates a strategy where the profits are reapplied to the market, facilitating the calculation of the total cumulative returns. At the end of each trading year, every position is closed.

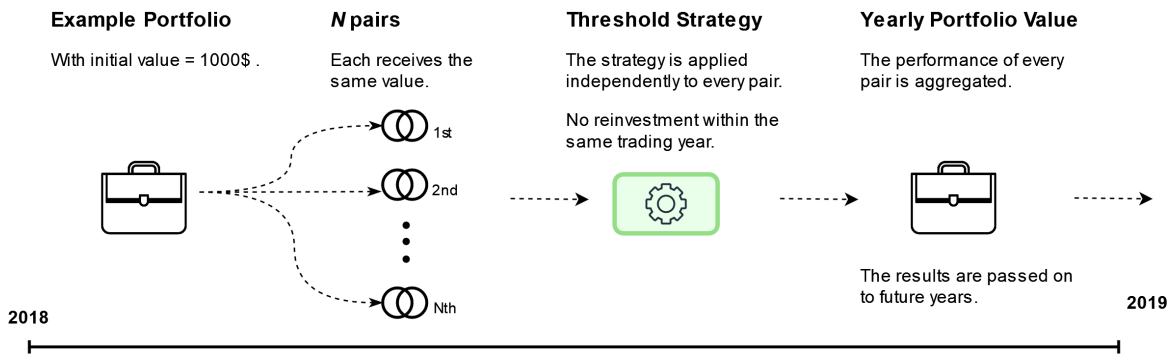


Figure 5.7: Portfolio management during the experiment.

Every pair in a portfolio is equally weighted, and the value designated to each pair, p_0, is simply the initial portfolio value for that year divided by the number of pairs. There is no reinvestment within the same trading year, in order to avoid losing in one big trade all the good work generated in a series of successful, consecutive small trades. Each component of the pair receives the same investment. The amount applied to each component at instant t, $p(t)$, is described by:

$$p(t) = min(p_0, p(t-1)). \tag{5.2}$$

5.6 Evaluation Metrics

This section presents the metrics used to assess the performance of the system. Regarding the forecasting algorithms, a relevant indicator is introduced, the "% of spread's directions correctly predicted".

5.6.1 Forecasting Performance

Measuring forecast accuracy is not a trivial task as there is no one-size-fits-all indicator. Several Key Performance Indicators (KPI) will be used in order to better assess the results of the forecasting algorithms, namely MAE, MSE and RMSE. These metrics are described in appendix A.

- **Direction of the spread**

The trading strategy applied relies mostly on the predicted trend and direction of the spread, rather than on its predicted value. Consider the case where the spread s at instant t has value 5, $s_t = 5$ and will rise in the next day to 10, $s_{t+1} = 10$. If model A predicts that the spread will grow to $s^{*(A)}_{t+1} = 100$, trading strategy A will long the spread. In the other hand, if model B forecasts that the spread will decrease, $s^{*(B)}_{t+1} = 4$, strategy B will short the spread. Although model B's forecast is closer to the observed value in s_{t+1}, as model A correctly guessed the spread's direction, it is strategy A that profits.

To further analyze the performance of the forecasting models, a new performance metric will be introduced, to compliment the information revealed by the KPIs and study how the ML models predict the direction of the spread. The metric "% Direction of the spread" reflects the percentage of directions correctly predicted.

5.6.2 Trading Performance

- Return On Investment

Return On Investment (ROI) is a standard, universal measure of profitability. It compares the gain or loss from an investment relative to its initial investment, and immediately reveals whether the venture was successful or not. Commonly expressed in percentage, ROI is given by

$$ROI = \frac{\text{Final Portfolio Value - initial Investment}}{\text{Initial Investment}} \times 100 \tag{5.3}$$

One disadvantage of ROI is that it doesn't account for how long an investment is held. As so, both the annualized and the total cumulative ROI will be calculated for each portfolio.

- Maximum Drawdown

Maximum Drawdown (MDD) is the maximum observed loss from a peak to a low point of a portfolio, before a new peak is attained. The value is computed with respect to the account balance during the trading period and is presented in percentage terms.

$$MDD = \frac{\text{Peak Value - Low Point Value}}{\text{Peak Value}} \times 100 \tag{5.4}$$

It serves as an indicator of downside risk over a specified time period and is useful to assess the relative riskiness of one investment strategy versus another, as it focuses on capital preservation, a key concern for most investors. For two given strategies that may have the same average performance, the one with the smallest MDD would be preferred, as this indicates that the losses were small at any given instant.

The MDD will be computed for each trading year and for the entirety of the trading period as well.

5.7 Conclusion

This chapter detailed the system implementation. Firstly, the data used was described, and the sliding window testing method was illustrated. Secondly, the architecture of the entire system was addressed,

combining the PF and the forecasting TS modules to address both research questions. Next, the hyperparameters selected for the NSGAs and the ML algorithms were justified. Finally, the back-testing features that add realism to the experiment were highlighted, along with the performance metrics that will be used to assess the results.

Table 5.3 encapsulates the hyperparameters selected for the algorithms.

Table 5.3: Hyperparameters for the algorithms employed.

NSGA II and III	
Population	50
Termination criterion	80 generations
Selection, Sampling, Crossover, Mutation	Tournament, Binary Random, Two Point, Bit Flip
Reference directions (NSGA III)	Riez s-Energy
NSGA II objective functions:	min. cointegration, max. volatility
NSGA III objective functions:	min. cointegration, max. volatility, max. NZC, min. half-life
Forecasting Models	
ARIMA	p = 5, d = 1, q = 0
XGBoost	n estimators = 400, η = 0.05, $\lambda = 1$, γ = 0.005, patience = 5

6

Results

Contents

6.1 Pair Formation Results 63

6.2 Forecasting-based Trading Stage 70

6.3 Forecasting trading results 72

This chapter presents and analyzes the results obtained. Firstly, the PF module is studied, concerning RS I. Then, the forecasting-based TS is examined, regarding RS II.

6.1 Pair Formation Results

The following section explores the formation of different types of pairs and their performance when traded from 2018 to 2021. The number of pairs clustered for each case study is presented in table 6.1.

Table 6.1: Number of pairs selected for the different types of pairs.

Formation	Cases				
Period	**SOUV**	**BOUV**	**BOMV**	**MOUV**	**MOMV**
'15-'17	21	4	27	7	8
'16-'18	36	4	17	7	11
'17-'19	61	6	23	10	12
'18-'20	25	2	38	9	14
Total	143	16	105	33	45

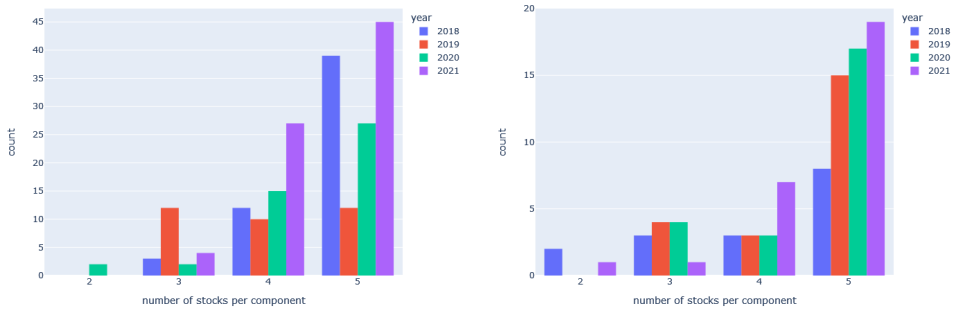
Given the combinatorial nature of multivariate pairs, it comes as no surprise that both the methods BOMV and MOMV select more pairs than their univariate equivalents. The number of pairs BOUV is the smallest by a substantial margin, which could be explained by the fact that having to minimize two functions (without having the advantage of multiple multivariate combinations) is a quite restrictive criterion. At last, the SOUV pairs, the ones formed with the traditional iterative approach, are the most numerous. As the stocks selected belong to the same financial sector, Real Estate, it is logical that a large portion of them are cointegrated, since they are subjected to the same economical stimuli, hence the bigger number of pairs. What is left to verify is if a larger number of pairs translates to more profitable portfolios.

BOMV and MOMV pairs are multivariate, having components composed of up to 5 stocks. The algorithm, through its natural selection filter, will determine whether more crowded pairs present more profit potential.

Fig. 6.1 displays the occurrences of the number of stocks per component. The majority of pairs "opt for" having more stocks per component, 4 to 5. Future sections will analyze if more populated pairs perform better.

6.1.1 NSGA II application

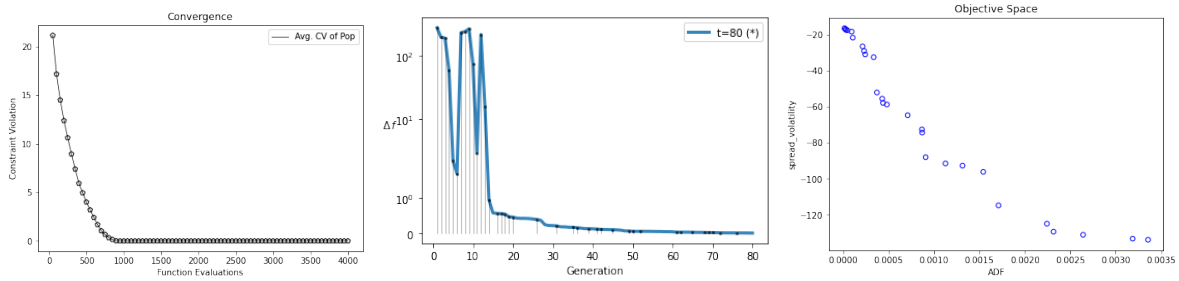
The following section highlights some details of the application of NSGA II.



(a) **(b)**

Figure 6.1: Number of stocks per component in BOMV (a) and MOMV (b) pairs.

As the algorithm iterates, its population adapts to the constraints imposed (or rather, the individuals that do not conform are eliminated). Fig. 6.2 (a) displays the evolution of the average constraint violation for BOMV pairs in 2018. In this case, almost every solution is feasible from generation 20 (or 1000 evaluations) onward.



(a) **(b)** **(c)**

Figure 6.2: Constraint violation (a), evolution of the objective functions (b) and selected pairs (c) for BOMV pairs in 2018.

Fig. 6.2 (b) displays the evolution of the objective space along generations. It is possible to verify that the change is minimal from the 50th generation onward, suggesting that terminating the algorithm at the 80th generation, as proposed in Chapter 5, is reasonable.

Finally, fig. 6.2 (c) shows the solutions selected in the objective space. BOMV pairs are formed by satisfying two objectives: the pair should be as cointegrated as possible, and its spread should be as volatile as possible. The disposition of the solutions allows to visualize a Pareto-optimal front.

Table 6.2: Trading Performance of the different types of pairs during 2018 to 2021.

Test Period	2018	2019	2020	2021	Average	Total
SOUV						
ROI	**2,38**	**-5,52**	**-3,95**	**1,57**	**-1,38**	**-5,64**
MDD	**4,27**	**8,07**	**10,08**	**3,30**	**6,43**	**15,59**
# pairs	21	36	61	25	35,75	143
% profitable pairs	76,19	41,67	49,18	52,00	54,76	51,75
% non-convergent pairs	66,67	63,89	72,13	76,00	69,67	69,93
# trades	70 - 33	83 - 51	167 - 93	73 - 34	98,25 - 52,75	393 - 211
BOUV						
ROI	**-1,11**	**-3,55**	**26,67**	**16,91**	**9,73**	**41,25**
MDD	**6,34**	**8,50**	**5,57**	**6,76**	**6,79**	**11,23**
# pairs	4	4	6	2	4,00	16
% profitable pairs	75,00	25,00	83,33	100,00	70,83	68,75
% non-convergent pairs	75,00	75,00	33,33	50,00	58,33	56,25
# trades	12 - 6	12 - 5	26 - 4	11 - 1	15,25 - 4	61 - 16
BOMV						
ROI	**2,71**	**-2,48**	**-14,31**	**-16,61**	**-7,67**	**-28,43**
MDD	**3,14**	**7,23**	**16,13**	**16,61**	**10,78**	**32,50**
# pairs	27	17	23	38	26,25	105
% profitable pairs	85,19	41,18	8,70	0,00	33,76	30,48
% non-convergent pairs	85,19	17,65	95,65	65,79	66,07	69,52
# trades	85 - 35	43 - 21	63 - 48	18 - 99	52,25 - 50,75	209 - 203
MOUV						
ROI	**0,42**	**-6,62**	**8,29**	**5,95**	**2,01**	**7,59**
MDD	**6,89**	**7,69**	**12,57**	**3,19**	**7,58**	**17,95**
# pairs	7	7	10	9	8,25	33
% profitable pairs	71,43	14,29	60,00	55,56	50,32	51,52
% non-convergent pairs	85,71	100,00	70,00	55,56	77,82	75,76
# trades	23 - 11	17 - 12	39 - 9	28 - 14	26,75 - 11,5	107 - 46
MOMV						
ROI	**4,21**	**-7,49**	**-13,11**	**-11,37**	**-6,94**	**-25,76**
MDD	**1,18**	**10,20**	**14,24**	**11,73**	**9,34**	**29,60**
# pairs	8	11	12	14	11,25	45
% profitable pairs	62,50	18,18	33,33	0,00	28,50	24,44
% non-convergent pairs	50,00	100,00	75,00	42,86	66,96	66,67
# trades	30 - 6	31 - 21	27 - 27	13 - 35	25,25 - 22,25	101 - 89

6.1.2 Trading Performance

Table 6.2 summarizes the performance of the different PF methods during each test period from 2018 to 2021.

Highlighted in bold are the most relevant performance metrics, ROI and MDD and, at the right most columns, the average and the cumulative results are presented. The cumulative/total value is particularly important since it exposes the performance of the portfolio throughout the entirety of the period - how much would an investor gain/lose if the corresponding case study was selected. Additionally, the number of pairs, the percentage of profitable and non-convergent pairs and the number of trades (profitable - unprofitable) give pertinent insight into the success/failure of the pair formation methods.

A quick glance to the ROI values reveals that the performance of most PF methods was rather disappointing. With the exception of BOUV and MOUV, all the portfolios resulted in losses. The BOUV pairs outperformed the others by a considerable margin, yielding $9,73\%$ yearly on average. This technique is clearly the superior one and will be further analyzed.

- Univariate vs. Multivariate pairs

By comparing the returns of the univariate portfolios with the multivariate ones, the research question "Are multivariate pairs better?" can be answered quite convincingly: no, multivariate pairs are no better, or are actually inferior to univariate pairs. Not only the returns are much worse, but also the MDD are larger. Naturally, the percentage of profitable pairs is inferior on multivariate pairs.

This conclusion corroborates the findings in [19]: univariate pairs present themselves as more profitable. The motives behind multivariate pairs' lackluster performance are examined in section 6.1.4.

- Forming pairs with MOO

Concerning the second research question, the verdict is not as evident. "Does MOO group better pairs?": yes, both BOUV and MOUV portfolios outperform the baseline SO approach. However, optimizing more objective functions is not synonym to better results, as BOUV proves to be better than MOUV. The BOUV and MOUV portfolios present a much smaller number of selected pairs, when compared with the other approaches, but the few pairs adopted present much higher profit potential. Furthermore, fewer pairs induce fewer trades, resulting, consequently, in smaller total transaction costs, as stated in [49].

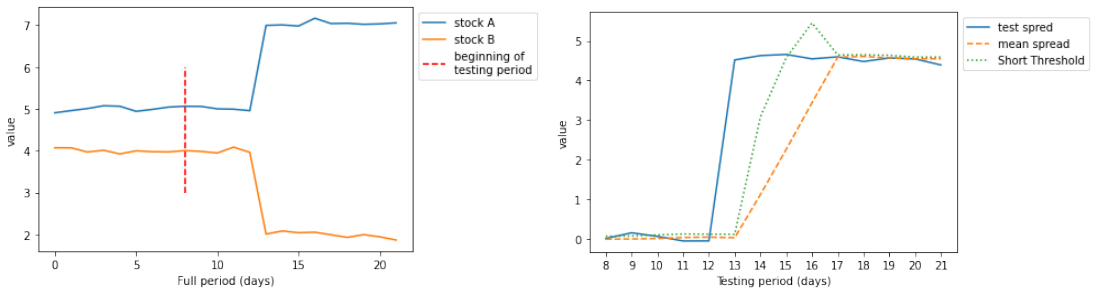
6.1.3 How does a trade result in a loss?

The poor results obtained in most portfolios justified a deeper analysis into the pairs' behavior. The following section delves into the motives, highlighted in bold, that negatively impact a pair's performance.

- **Pairs alter their behavior**

Consider the following two signals depicted in fig. 6.3. Although oversimplified, this example will serve to demonstrate the influence of ”bad behaved” pairs and the resulting consequences in profitability. The formation period lasts eight days, and the beginning of the testing period is signaled by the vertical dashed line. The parameters μ, σ of the threshold trading strategy will be calculated based on the last four days.

The hypothetical stock prices follow each other closely, but in the testing period their behavior is extremely different. Not only do they diverge but also stay apart for a considerable amount of time.



(a) **(b)**

Figure 6.3: Example of pair (a) and spread (b) that result in a loss.

Fig. 6.3 (b) displays the evolution of the pair's spread during the testing period. Reflecting the behavior of the stocks, the spread increases greatly and then stabilizes in a considerable distance. Around the 12th day, the spread surpasses the green Short Threshold, thus a market position is entered. Remember that, when the spread is bigger than this threshold, the trader believes that it will decrease, hence the name ”Short Threshold”. Unfortunately, the pair will not converge, and its spread will not decline.

Now, the culprit of the subsequent unprofitable trade is the spread's mean, μ. As the spread increases, its mean (calculated with the observations from the past 4 days), rises as well. At some point, around day 17, the spread touches the mean, and the trading rules dictate that the trade is closed, as it is assumed that the pair has converged. However, the reality is quite the opposite. The pairs diverged for so long that the strategy believes that their now distant relationship is the new normal.

In conclusion, if a pair behaves very differently in the testing period, which happens frequently, a trade could result in a loss.

- **Non convergent pairs**

A further reason behind fruitless trades are non-convergent pairs. A pair is considered non-convergent if, when the trading period ends, it still presents an open market position. The position is then forcedly

closed, and the outcome is usually negative, since the pair has not converged yet.

6.1.4 Worst performing portfolios

Having presented the factors responsible for unprofitable trades, it will now be tested if they are present in the worst performing portfolios, SOUV, BOMV, MOMV.

To verify the hypothesis that pairs alter their relationship's character in the test period, the number of pairs that still are cointegrated was counted. Remember that cointegration reveals a long-lasting proximity relationship, and that all types of pairs in the different case studies were formed concerning the cointegration $pvalue$ (among other objectives, of course, for the bi and many-objective pairs).

Table 6.3: Number of pairs still cointegrated over the total number of pairs.

Testing Year	**SOUV**	**BOUV**	**BOMV**	**MOUV**	**MOMV**
2018	1/21	0/4	0/27	0/7	0/8
2019	5/36	1/4	0/17	1/7	0/11
2020	13/61	1/6	0/23	3/10	0/12
2021	1/25	0/2	0/38	0/9	0/14

Table 6.3 displays the number of pairs that are cointegrated in the testing period, over the total number of pairs. It is possible to observe that the vast majority of pairs, no matter the formation technique used, are not cointegrated, which raises the hypothesis that cointegration is not the best criterion to form pairs [50]. This is true even for SOUV pairs, that rely solely on cointegration to form pairs. It is evident that pairs change their behavior in the testing period - the close relationship during the formation period does not persist in testing - proving to be a determining factor of unprofitability. Fig. 6.4 illustrates the change in the behavior of the pair EQR-DLR from the training to the testing period.

The worst results are observed in the multivariate pairs: 0% for both BOMV and MOMV. The bigger the number of stocks held, the higher the entropy, the more prejudicial are the differences in behavior from each individual stock that composes the pair. This phenomenon explains the mediocre performance from the multivariate pairs, and further supports the conclusion that multivariate pairs are no better than univariate ones.

Summarizing, even if a pattern is observed for a long time in the past, there is no guarantee that the stocks will obey such balance in the testing period. The saying "the past does not predict the future" is specially applicable to a PT strategy.

An additional negative agent is the sizeable number of non-convergent pairs. The percentage of non-convergent pairs is significant across all types, with the smallest average percentage being $58,33\%$ for BOUV pairs (not surprisingly, the best performing overall), and the biggest $78,82\%$ for MOUV. But how prejudicial is the non-convergence?

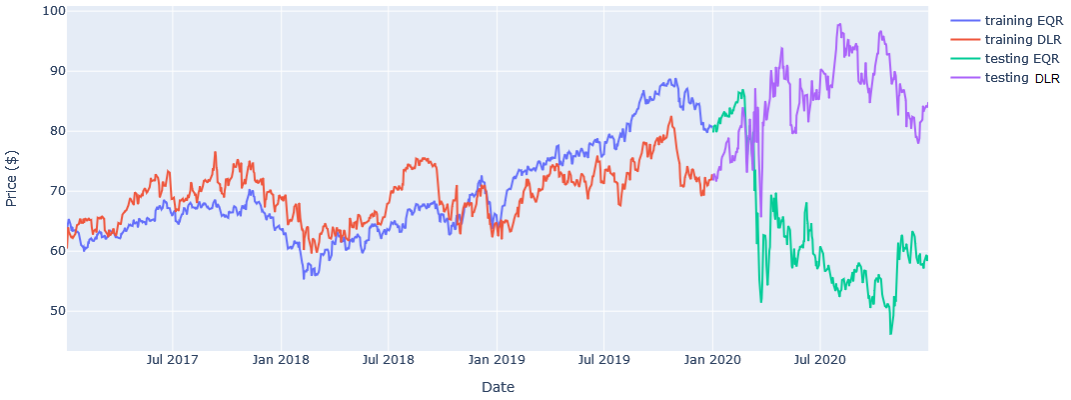
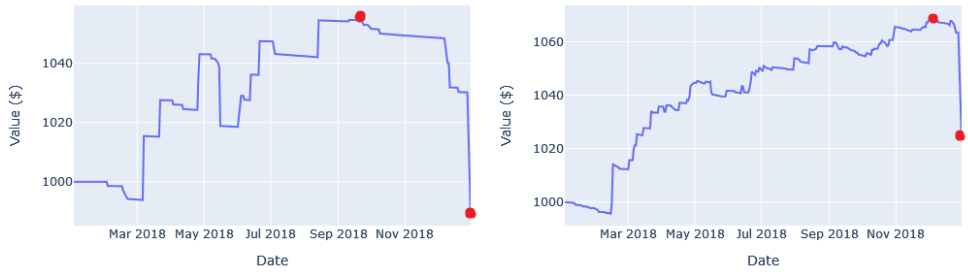


Figure 6.4: Different behavior of EQR-DLR in the training and testing periods.

Fig. 6.5 displays the evolution of portfolios BOUV, SOUV in 2018, where the red dots signal the MDD. It is possible to observe that an otherwise positive year is quickly turned red due to the massive negative spike caused by the forced exit of non-convergent positions, in the last day of the trading year. This observation endorses the conjecture that non-convergent pairs are highly (negatively) influential in the outcome of the strategy.



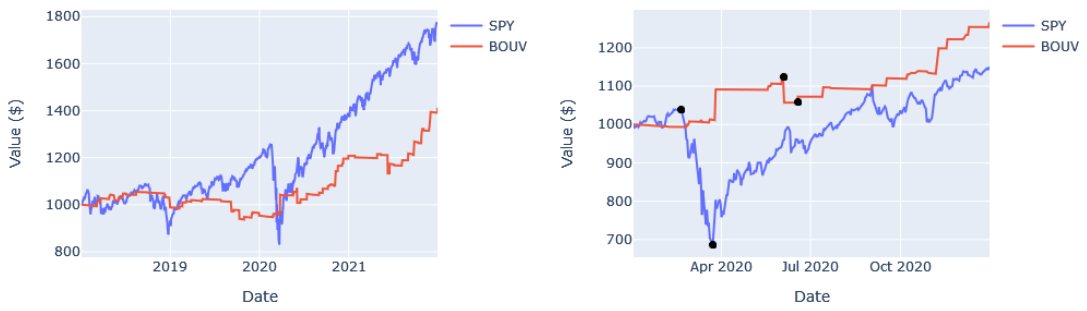
(a) **(b)**

Figure 6.5: Influence of non-convergence in BOUV (a) and SOUV (b) in 2018.

It is hypothesized that limiting trades in the later part of the testing period could minimize the effect of non-convergent pairs and improve the strategy.

6.1.5 Best performing portfolio - BOUV

The BOUV pairs distinguished themselves by being the most profitable by quite some margin. In fact, the BOUV portfolio outperforms the baseline SOUV, and its study leads to some noteworthy observations.



(a) **(b)**

Figure 6.6: Comparison of SPY and BOUV portfolios in 2018-2021 (a), and during 2020 only (b).

Fig. 6.6 (a) compares the BOUV portfolio with the SPY index, commonly used as a benchmark, from 2018 to 2021. Although SPY bests the BOUV portfolio over the entire period, the PT strategy performs better in 2020, not only having a bigger ROI but also a lower MDD, signaled by the black dots (fig. 6.6 (b)). Given that 2020 was characterized by the March market crash, one of the most significant in the last decade and consequence of the COVID-19 pandemic, the fact that BOUV outperforms SPY in 2020 confirms the idea found in the literature that PT is robust to market crashes.

6.2 Forecasting-based Trading Stage

The following section focuses on the results of the forecasting-based trading strategy, in an effort to surpass the traditional threshold-based strategy. The forecasting and trading performances of the ARIMA and XGBoost models will be analyzed.

The forecast-based module uses the BOUV pairs, since it is desirable to compare this alternative strategy with the best performing pairs of the traditional method.

6.2.1 Forecasting performance

Table 6.4 summarizes the results for the prediction of all pairs considered in 2018, 2019. The metric "percentage of spread's directions correctly predicted" is highlighted as it is relevant to explain the performance of the algorithms in a trading simulation, explored in a future section. The "A" or "X" in the model row stands for ARIMA or XGBoost. The results presented were obtained with the configurations described in Chapter 5.

By analyzing the results from table 6.4, it is evident that the ARIMA model performs better than the XGBoost at predicting the spread's future value. This result is particularly curious given that the ARIMA model is much simpler in nature, and much older as well. One would expect that the more recent and

Table 6.4: Forecasting performance of ARIMA and XGBoost in 2018 and 2019.

Test Year	**2018**									
Pair	DLR-AVB		KIM-ESS		DLR-UDR		EQIX-AVB		**Average**	
Model	A	X	A	X	A	X	A	X	A	X
Test Score										
RMSE	1,30	1,78	0,28	0,45	1,28	1,65	5,44	5,88	2,07	2,44
MSE	1,68	3,17	0,08	0,20	1,64	2,71	29,66	34,59	8,26	10,17
MAE	0,88	1,22	0,22	0,30	0,86	1,07	3,83	4,11	1,45	1,68
Direction %	28,4	59,60	21,20	53,60	30,40	60,00	44,40	66,40	**31,10**	**59,9**
Test Year	**2019**									
Pair	FRT-EQR		SBAC-EQR		MAA-EQR		ESS-DLR		**Average**	
Model	A	X	A	X	A	X	A	X	A	X
Test Score										
RMSE	1,08	1,64	2,56	3,58	0,68	0,82	3,15	5,55	1,87	2,90
MSE	1,17	2,70	6,56	12,79	0,46	0,68	9,95	30,75	4,53	11,73
MAE	0,80	1,14	1,75	2,34	0,49	0,59	2,38	3,42	1,35	1,87
Direction %	28,40	54,8	36,00	59,2	27,20	56,4	39,20	64,4	**32,70**	**58,7**

complex model XGBoost would outshine the simpler model. This dissonance is aggravated by the fact that the ARIMA is much quicker to fit.

The results justify a deeper look into the literature, in search of similar cases where the ARIMA model performs the best. In [24], an ARIMA model yields better results than both a LSTM and a LSTM Encoder-Decoder. This suggests that the extra complexity added by the XGBoost might be unnecessary if the objective is to predict a signal as close to the real one as possible.

However, the KPIs do not reveal entirely the quality of the forecasts. Fig. 6.7, illustrates the results of the forecasting of the KIM-ESS pair. It is possible to observe that, while the ARIMA model does indeed follow the real price more closely, it does so in an almost lagged fashion. The predicted curve follows the movement of the real spread with the lag of one day. This behavior could be compared to a persistent forecasting method, that sets tomorrow's value equal to today's.

$$\hat{s}_{t+1} = s_t \tag{6.1}$$

Revisiting fig. 6.7, notice how the ARIMA signal is the real, blue, spread, shifted one day to the right.

On the other hand, the spread predicted by the XGBoost almost resembles a succession of step signals, due to the fact that the XGBoost model is fitted each 5 days. Consequently, during the five-day

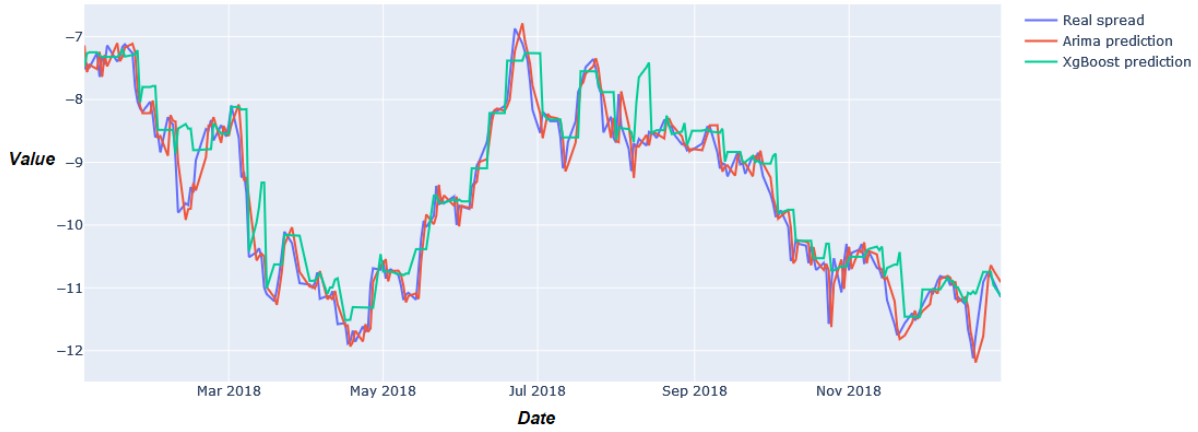


Figure 6.7: XGBoost and ARIMA forecasts for KIM-ESS in 2018.

period when the model is not re-trained, the XGBoost algorithm does not present the ability to closely follow the original signal. While not being as close to the real spread as the ARIMA, it shows a somehow more ”independent” behavior, anticipating the spread's direction of movement instead of lagging behind. Finally, while the ARIMA outperforms the XGBoost in the traditional KPIs, the latter model has much better results when predicting the future direction of the spread. These two different behaviors will largely influence the outcome of the trading strategies, as it will be discussed in the following section.

6.3 Forecasting trading results

The current section presents the trading results of the forecasting strategy, that will allow to determine the best algorithm. Furthermore, by comparing the forecasting-based TS with the results of the threshold-based strategy, the research question ”Does a ML based strategy outperform the traditional strategy?” will be solved.

Table 6.5 synthesizes the results of the Forecasting-based strategy in 2018 and 2019. As a term of comparison, the best performing pairs, BOUV, are presented for the same period, employing the traditional threshold-based strategy. Besides the most relevant metrics, ROI and MDD, the percentage of profitable pairs and the number of positive and negative trades are displayed.

XGBoost obtains the best trading results by a large margin, yielding $49,24\%$ of profit over the 2 years, while the ARIMA portfolio presents a loss. Furthermore, it also outperforms the BOUV pairs, leading to believe that ML can indeed improve the traditional strategy. The XGBoost portfolio not only produces the best overall ROI, but also presents the minimum MDD, meaning that the portfolio does not decrease

Table 6.5: Forecasting-based strategy performance in 2018 and 2019.

Test Period	**2018**	**2019**	**Average**	**Total**
ARIMA				
ROI	**-28,05**	**-40,44**	**-34,25**	**-57,15**
MDD	**28,77**	**40,73**	**34,75**	**57,36**
# pairs	4	4	4	8
% profitable pairs	25	0	12,5	12,5
# trades	177 - 302	192-392	184,5 - 347	369-694
XgBoost				
ROI	**43,84**	**3,75**	**23,80**	**49,24**
MDD	**4,84**	**5,36**	**5,10**	**5,36**
# pairs	4	4	4	8
% profitable pairs	100	50	75	75
# trades	282-144	275-146	278,5 - 145	557 - 347
BOUV				
ROI	**-1,11**	**-3,55**	**-2,33**	**-4,62**
MDD	**6,34**	**8,50**	**7,42**	**8,50**
# pairs	4	4	4	8
% profitable pairs	75	25	50	50
# trades	12-6	12-6	12-5,5	24-11

drastically in value at any given time. In fact, the biggest downtrend is only 5,36%, observed in 2019. Fig. 6.8 depicts the evolution of the portfolios compared with the SP500. The MDD of the XGBoost portfolio is marked with black dots.

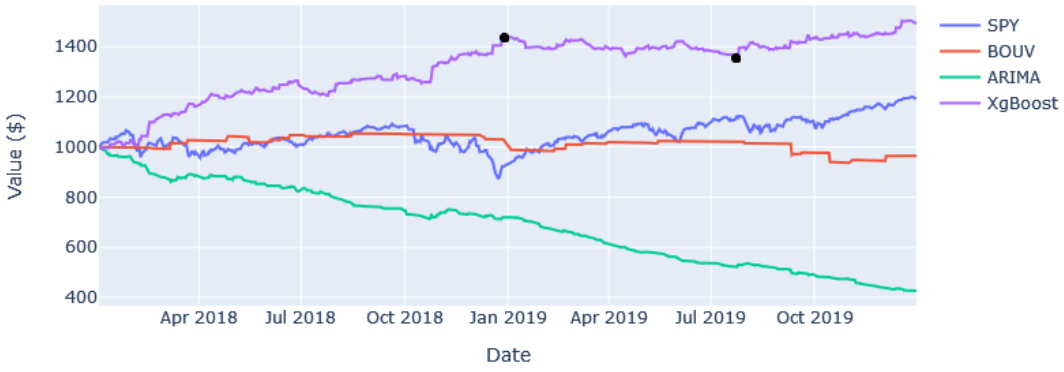


Figure 6.8: Xgboost, ARIMA and BOUV portfolios compared with the SP500 in 2018-2019.

6.3.1 Why does the ARIMA portfolio perform so poorly?

The following example demonstrates how the lagged behavior of the ARIMA predictions negatively impacts the trading performance. A trading execution will be demonstrated for the KIM-ESS pair during the last days of June.

Fig. 6.9 displays the ARIMA forecast for the pair KIM-ESS during the end of June of 2018. The spread value being negative does not constitute an issue to the strategy, it merely means that ESS is bigger in value than KIM in this period, since the spread is defined as $s = KIM - \beta \cdot ESS$. Table 6.6 presents the predicted movement, the real spread's direction, and the action executed by the trader. Recall that the trader only acts when there is an inversion in the trend. The actions are shadowed in either green or red depending on whether they result in profit or loss, respectively.

Table 6.6: Exemplifying the forecasting strategy for ARIMA.

Day	19	20	21	22	25	26	27	28
Real direction	Up	Up	Up	Down	Down	Down	Down	Down
Predicted direction	Down	Down	Up	Down	Up	Up	Down	Up
Trader action	Short		Long	Short	Long		Short	Long

The lagged behavior of the prediction is clearly visible in fig 6.9. The ARIMA model does not antici-

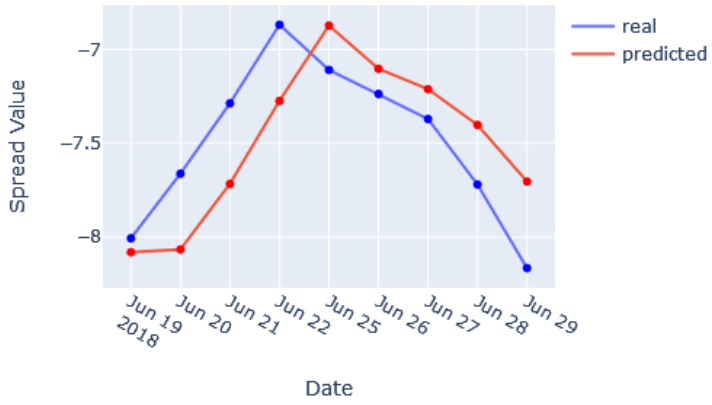


Figure 6.9: ARIMA forecast for KIM-ESS in June 2018.

pate the spread, simply deviates marginally from the observed value and, in the majority of times, does so in terrible fashion: to the ARIMA, the spread either goes up or down, offering no better prediction than a coin toss. This rather strong conclusion is supported by the metric "% percentage of directions correctly predicted", lower than 50%, as presented in table 6.4. This leads to incorrectly predicted trends, displayed in table 6.6. Consequently, as the value of the incorrect decisions outweighs the profit by the correct ones, the return in this example is negative.

The low percentage of correctly guessed directions has an additional implication: as the forecasting strategy is based on the predicted trend, the consecutive bad predictions compound into a disastrous strategy, justifying the substandard performance of the ARIMA portfolio and the considerable fraction of negative trades.

6.3.2 XGBoost: best overall portfolio - Comparison with related works

The XGBoost portfolio nets the best results, not only for the ML models but also including the portfolios studied in the PF module. The high percentage of directions correctly predicted results in a majority of positive trades and cumulative profit.

The contrast in profitability from 2018 to 2019 is explained by analyzing the return for each trade. Although the ratio of positive/negative trades almost does not change in these two years, the negative trades are much more damaging. In effect, unprofitable trades are bigger in absolute value than profitable ones. As consequence, a major unsuccessful action invalidates the gains obtained in a series of small profitable trades.

The positive performance of the XGBoost portfolio answers the research question concerning the Forecasting-based Trading Stage: ML does indeed contribute positively to a PT strategy, surpassing the threshold-based TS for the same pairs. However, it must be noticed that 2018 and 2019 are precisely the

worst years for the BOUV pairs in the threshold-based strategy. Furthermore, the experiment performed should encompass a larger period to further validate this conclusion.

But how does the XGBoost portfolio fare against the benchmark results enumerated in section 2.6? With an average return of 23,8%, the developed forecasting-based strategy outclasses [1, 7], that use non-CI techniques. In regards to CI-based strategies, this work outperforms the likes of [19, 24]. Nevertheless, the returns obtained fall short of the ones in [16], that achieves 33% yearly returns, with the caveat that it does not consider transaction costs.

7

Conclusions and Future Work

Contents

7.1 Conclusions 79

7.2 Future Work 80

7.1 Conclusions

The current thesis intended to improve a Pairs Trading investment strategy with the application of Machine Learning techniques. Having exposed the results obtained, it is now possible to comprehensively address the research questions that led this work.

The question ”Are multivariate pairs better than univariate ones?” is revisited first. Multivariate pairs were introduced as more appealing, based on the rationale that diversifying and increasing the assets in hand would reduce volatility and produce better quality pairs. In a PT strategy, however, this could not be farther from the truth. Multivariate pairs were found to be the worst performing. In fact, it was revealed that most pairs change their behavior from the formation to the testing period. In multivariate pairs, this phenomenon is aggravated by the higher number of stocks, decreasing the quality of the pair. In conclusion, the results support quite compellingly that multivariate pairs are worse than univariate.

The pairs formed by satisfying multiple objectives, BOUV and MOUV, outperform the standard iterative approach of forming pairs. Hence, the research question ”Does MOO find more profitable pairs?” has an affirmative answer: yes, forming pairs as a MOOP is beneficial. Yet, the results reveal some nuances: simply adding more objective functions does not yield better results. In fact, the experiment makes clear that the pairs created by satisfying two objectives (bi-objective) are the best performing.

Finally, this work attempts to improve the trading stage by applying a forecasting-based strategy. The XGBoost portfolio beats the traditional trading strategy, achieving 23,8% average returns and exhibiting short declines, with a MDD of 5,4%. Additionally, the results obtained outperform most of the works found in the literature, most importantly, the ones that do not apply computational intelligence, accomplishing a major goal of this thesis.

Overall, the trading system developed beats the market in 2018 and 2019. The positive results obtained, both in the pair formation stage and in the trading stage, allow to conclude that this thesis succeeded in improving a Pairs Trading Portfolio by applying Machine Learning techniques.

Some additional contributions of this thesis are:

- Emphasizing the importance of using only past values in the normalization of the spread, and the influence of the number of past days considered.
- Finding that most pairs break their cointegration relationship in the trading period.
- Evidencing the negative influence of non-convergent pairs.
- Testing the strategy during a major market collapse, demonstrating that it is robust to stock market crashes.

7.2 Future Work

This study touched several very interesting topics that could justify further investigation. Some possible future directions of research are:

- Study the influence of the formation and trading period duration in the number of pairs cointegrated in the testing period.
- Develop a trading strategy that avoids non-convergent pairs by limiting trades in the later part of the trading period.
- Formulate a trend-based trading strategy as a classification problem.
- Diversify a PT-based strategy by forming pairs for each economic sector of the S&P 500.

Bibliography

[1] E. Gatev, W. N. Goetzmann, and K. G. Rouwenhorst, "Pairs trading: Performance of a relative-value arbitrage rule," *Review of Financial Studies*, vol. 19, pp. 797–827, 9 2006.

[2] A. Flori and D. Regoli, "Revealing pairs-trading opportunities with long short-term memory networks," *European Journal of Operational Research*, 2021.

[3] M. Fil and L. Kristoufek, "Pairs trading in cryptocurrency markets." [Online]. Available: https://www.binance.com/en

[4] C. Krauss, "Statistical arbitrage pairs trading strategies: Review and outlook," *IWQW Discussion Papers*, 8 2015. [Online]. Available: http://hdl.handle.net/10419/116783www.econstor.eu

[5] H. Chen, S. Chen, Z. Chen, F. Li, and P. of Accounting, "Empirical investigation of an equity pairs trading strategy," 2017.

[6] Y. Mai and S. Wang, "Whether stock market structure will influence the outcome of pure statistical pairs trading?" vol. 3, 2011, pp. 291–294.

[7] B. Do and R. Faff, "Does simple pairs trading still work?" *Financial Analysts Journal*, vol. 66, pp. 83–95, 7 2010.

[8] G. Shobana and K. Umamaheswari, "Forecasting by machine learning techniques and econometrics: A review." Institute of Electrical and Electronics Engineers Inc., 1 2021, pp. 1010–1016.

[9] J. Aldrich, "Correlations genuine and spurious in pearson and yule," *Statistical Science*, vol. 10, pp. 364–376, 1995.

[10] T. Vigen. Spurious correlations. [Online]. Available: https://tylervigen.com/spurious-correlations

[11] B. E. Sørensen, "Cointegration," 1992.

[12] E. P. Chan, "Algorithmic trading : winning strategies and their rationale," p. 207, 2013.

[13] M. C. Blázquez, C. D. la Orden De la Cruz, and C. P. Román, "Pairs trading techniques: An empirical contrast," *European Research on Management and Business Economics*, vol. 24, pp. 160–167, 9 2018. [Online]. Available: https://doi.org/10.1016/j.iedeen.2018.05.002

[14] L. Zhang, "Pair trading with machine learning strategy in china stock market." Association for Computing Machinery, 5 2021.

[15] C. F. Huang, C. J. Hsu, C. C. Chen, B. R. Chang, and C. A. Li, "An intelligent model for pairs trading using genetic algorithms," *Computational Intelligence and Neuroscience*, vol. 2015, 2015.

[16] A. Brim, "Deep reinforcement learning pairs trading with a double deep q-network." Institute of Electrical and Electronics Engineers Inc., 1 2020, pp. 222–227.

[17] C. Wang, P. Sandas, and P. Beling, "Improving pairs trading strategies via reinforcement learning." Institute of Electrical and Electronics Engineers Inc., 5 2021.

[18] M. S. Perlin, "M of a kind: A multivariate approach at pairs trading," *ICMA/Reading University*, 2007.

[19] J. Goldkamp and M. Dehghanimohammadabadi, "Evolutionary multi-objective optimization for multivariate pairs trading," *Expert Systems with Applications*, vol. 135, pp. 113–128, 11 2019.

[20] D. Dickey and W. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *Journal of the American Statistical Association*, vol. 74, p. 427, 1979.

[21] R. F. Engle and . C. W. J. Granger, "Co-integration and error correction: Representation, estimation, and testing," *Econometrica*, vol. 55, pp. 251–276, 1987.

[22] G. Vidyamurthy, "Pairs trading : quantitative methods and analysis," 2004. [Online]. Available: www.WileyFinance.com.

[23] J. F. Caldeira and G. V. Moura, "Selection of a portfolio of pairs based on cointegration: A statistical arbitrage strategy," *Brazilian Review of Finance*, vol. 11, pp. 49–80, 2013. [Online]. Available: https://ideas.repec.org/a/brf/journl/v11y2013i1p49-80.html

[24] S. M. Sarmento and N. Horta, "Enhancing a pairs trading strategy with the application of machine learning," 2020. [Online]. Available: https://doi.org/10.1016/j.eswa.2020.113490

[25] N. Huck, "Pairs selection and outranking: An application to the sp 100 index," *European Journal of Operational Research*, vol. 196, pp. 819–825, 7 2009.

[26] ——, "Pairs trading and outranking: The multi-step-ahead forecasting case," 2010.

[27] T. Kim, H. Y. Kim, and B. M. Tabak, "Optimizing the pairs-trading strategy using deep reinforcement learning with trading and stop-loss boundaries," *Complexity*, vol. 2019, 2019.

[28] J. P. Nóbrega and A. L. Oliveira, "A combination forecasting model using machine learning and kalman filter for statistical arbitrage," vol. 2014-January. Institute of Electrical and Electronics Engineers Inc., 2014, pp. 1294–1299.

[29] G. Rudolph, "Convergence of evolutionary algorithms in general search spaces," *Proceedings of the IEEE Conference on Evolutionary Computation*, pp. 50–54, 1996.

[30] K. Deb, R. E. Steuer, and R. Aj, "Bi-objective portfolio optimization using a customized hybrid nsga-ii procedure," 2011.

[31] M. Kaucic, M. Moradi, and M. Mirzazadeh, "Portfolio optimization by improved nsga-ii and spea 2 based on different risk measures," *Financial Innovation*, vol. 5, 12 2019.

[32] M. Awad, M. Abouhawwash, and H. N. Agiza, "On nsga-ii and nsga-iii in portfolio management," 2021.

[33] C. Bennett and M. A. Gil, "Equity derivatives europe measuring historical volatility," 2012.

[34] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, 4 2002.

[35] M. Pilát, "Evolutionary multiobjective optimization: A short survey of the state-of-the-art," 2010.

[36] M. T. Emmerich and A. H. Deutz, "A tutorial on multiobjective optimization: fundamentals and evolutionary methods," *Natural Computing*, vol. 17, pp. 585–609, 9 2018. [Online]. Available: https://link.springer.com/article/10.1007/s11047-018-9685-y

[37] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part i: Solving problems with box constraints," 2013. [Online]. Available: http://www.egr.msu.edu/

[38] J. Blank, K. Deb, Y. Dhebar, S. Bandaru, and H. Seada, "Generating well-spaced points on a unit simplex for evolutionary many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, pp. 48–60, 2 2021.

[39] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Applied Soft Computing Journal*, vol. 90, 5 2020.

[40] M. F. Anaghi and Y. Norouzi, "A model for stock price forecasting based on arma systems," 2012, pp. 265–268.

[41] K. W. Sin, H. H. Sheng, W. C. Zhi, L. P. Ling, and C. Dass, “Forecasting stock price using arma model,” vol. 2020, p. 59, 2020.

[42] S. Meisenbacher, M. Turowski, K. Phipps, M. Rätz, . D. Müller, V. Hagenmeyer, and R. Mikut, “Review of automated time series forecasting pipelines,” 2022. [Online]. Available: https://doi.org/10.1002/widm.1475

[43] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-August-2016, pp. 785–794, 3 2016. [Online]. Available: https://arxiv.org/abs/1603.02754v3

[44] C. Krauss, X. A. Do, and N. Huck, “Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the sp 500,” *European Journal of Operational Research*, vol. 259, pp. 689–702, 6 2017.

[45] J. Blank and K. Deb, “pymoo: Multi-objective optimization in python,” 1 2020. [Online]. Available: http://arxiv.org/abs/2002.04504http://dx.doi.org/10.1109/ACCESS.2020.2990567

[46] ——, “A running performance metric and termination criterion for evaluating evolutionary multi- and many-objective optimization algorithms,” 7 2020.

[47] B. Do and R. Faff, “Are pairs trading profits robust to trading costs?” 2011. [Online]. Available: http://ssrn.com/abstract=1707125

[48] M. Kociński, “On transaction costs in stock trading,” *Metody Ilościowe w Badaniach Ekonomicznych*, vol. 18, 4 2022. [Online]. Available: https://qme.sggw.edu.pl/article/view/2968

[49] M. J. Barclay, E. Kandel, and L. M. Marx, “The effects of transaction costs on stock prices and trading volume*,” *JOURNAL OF FINANCIAL INTERMEDIATION*, vol. 7, pp. 130–150, 1998.

[50] M. Clegg and C. Krauss, “Pairs trading with partial cointegration,” *Quantitative Finance*, vol. 18, pp. 121–138, 1 2018.

A

Key Performance Indicators

Next are described the metrics used to assess the forecasting performance of ARIMA and XGBoost.

- **MAE**

The MAE is one of the most commonly used KPI. As implied by its name, it is computed as the mean of the absolute error:

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t|, \tag{A.1}$$

where n is the total length of the period forecasted and $\hat{y}_t$ is the prediction at time t. One drawback of MAE is the fact that it is not scaled: if the MAE is 10 for a particular example, it is not possible to immediately evaluate the quality of the forecast. As such MAE values will be compared for the same pair.

- **MSE**

The MSE is the average squared distance between the observed and predicted values. However, since it uses squared units rather than the natural data units, the interpretation is less intuitive. It is computed

as:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2. \tag{A.2}$$

Squaring the error imposes that the MSE is greater or equal to zero, and emphasizes the influence of larger errors. This metric is then particularly useful to evaluate whether the forecasting model has large errors. Once again, the MSE is not scaled to the data, meaning that it is not possible to compare this metric across time series with different scales.

- **RMSE**

The RMSE is, naturally, related to the MSE. Given by

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}, \tag{A.3}$$

the square root relates this metric to the original data units. Just like the MSE, it penalizes large errors outliers, but while the MSE is analogous to the variance, the RMSE is akin to the standard deviation.

B

Parameters selection

The following appendix presents the experiments that justified the selection of some parameters in chapter 5.

In particular, the following implementation choices will be addressed:

- Why predict the spread and not predict two stocks individually?
- Normalization period for μ, σ when normalizing the spread.
- Termination criterion for the NSGAs.

B.1 Why predict the spread directly?

This work opted for forecasting the spread: based on past observations of the spread signal, predict the day-ahead value.

Another approach could be used: predict each of the stocks that compose the pair and then obtain the predicted spread as the difference between the predicted stocks.

Table B.1 encapsulates the performance of both approaches, where s stands for "spread" and $2s$ for "two stocks". It is possible to observe that predicting the spread directly obtains better results. Additionally, forecasting the spread instead of two signals has the added benefit of taking only half of the computational time.

Table B.1: Comparison of predicting the spread directly or the 2 stocks individually .

Test Year	2018									
Pair	DLR-AVB		KIM-ESS		DLR-UDR		EQIX-AVB		Average	
Predicted Signal	s	2s	s	2s	s	2s	s	2s	s	2s
Test Score										
RMSE	1,779	1,863	0,446	0,456	1,647	1,914	5,881	6,785	**2,438**	**2,755**
MSE	3,165	3,472	0,199	0,208	2,713	3,664	34,587	46,034	**10,166**	**13,345**
MAE	1,22	1,259	0,301	0,319	1,068	1,311	4,113	4,831	**1,676**	**1,93**
Test Year	2019									
Pair	FRT-EQR		SBAC-EQR		MAA-EQR		ESS-DLR		Average	
Predicted Signal	s	2s	s	2s	s	2s	s	2s	s	2s
Test Score										
RMSE	1,642	1,771	3,576	3,818	0,822	1,115	5,545	5,241	**2,896**	**2,986**
MSE	2,696	3,137	12,786	14,574	0,675	1,242	30,751	27,478	**11,757**	**11,608**
MAE	1,136	1,266	2,34	2,662	0,594	0,779	3,423	3,816	**1,873**	**2,116**

B.2 Normalization Period - 63 days

This work emphasizes the importance of only using past values in the normalization of the spread (in the traditional threshold strategy).

The value selected for the number of days was 63, as it yielded the best experimental results.

Table B.2 displays the cumulative results of the threshold trading strategy for various normalization periods. Positive returns are highlighted in green.

Moreover, notice how the BOUV pairs are the best performing for every normalization period tested, further supporting the conclusion that these are the best type of pairs.

Table B.2: Trading results for different normalization periods from 2018 to 2021.

Normalization Period	256	126	**63**	36	15
Type of Pair					
SOUV	-5,09	-8,39	**-5,64**	-6,61	-14,75
BOUV	3,55	2,27	**41,25**	25,94	36,12
BOMV	-24,43	-34,04	**-34,35**	-36,97	-32,48
MOUV	-2,32	-7,91	**7,59**	-4,22	-10,57
MOMV	-14,22	-28,09	**-29,02**	-27,39	-16,25

B.3 NSGA Termination Criterion - 80 generations

The reasons that motivated the restriction of the training process to 80 generations were:

- Reduce the training time to a tolerable amount (less than 1 and a half hour), allowing to explore several configurations and types of pairs;
- The objective space stabilizes before generation 80.

The termination criterion used to halt the training of the NSGA algorithms was a max number of generations, 80 generations. Naturally, the ideal termination point would be obtained by monitoring the objective space and stopping the training process when no improvements have been made for some generations.

However, the ideal stopping criterion requires a much bigger training effort and computational power. In order to test as many hypotheses and configurations as possible, it was decided that each training session would be restricted to a tolerable amount of time. Consequently, a compromise has been made: the results might not be the best possible, but will allow to answer the multiple research questions and break as much ground as possible.

Fig. B.1 depicts the evolution of the objective space during 2019, for all types of pairs formed with NSGA. The stopping point of 80 generations is considered appropriate as, by observing each example in fig. B.1, the objective space stabilizes or does not change well before the 80 generation mark.

Finally, the training of the NSGA algorithms with the adopted configuration of max gen = 80, lasted for approximately 1 hour and 20 minutes in a *google colab* environment (20 minutes per type of pair).

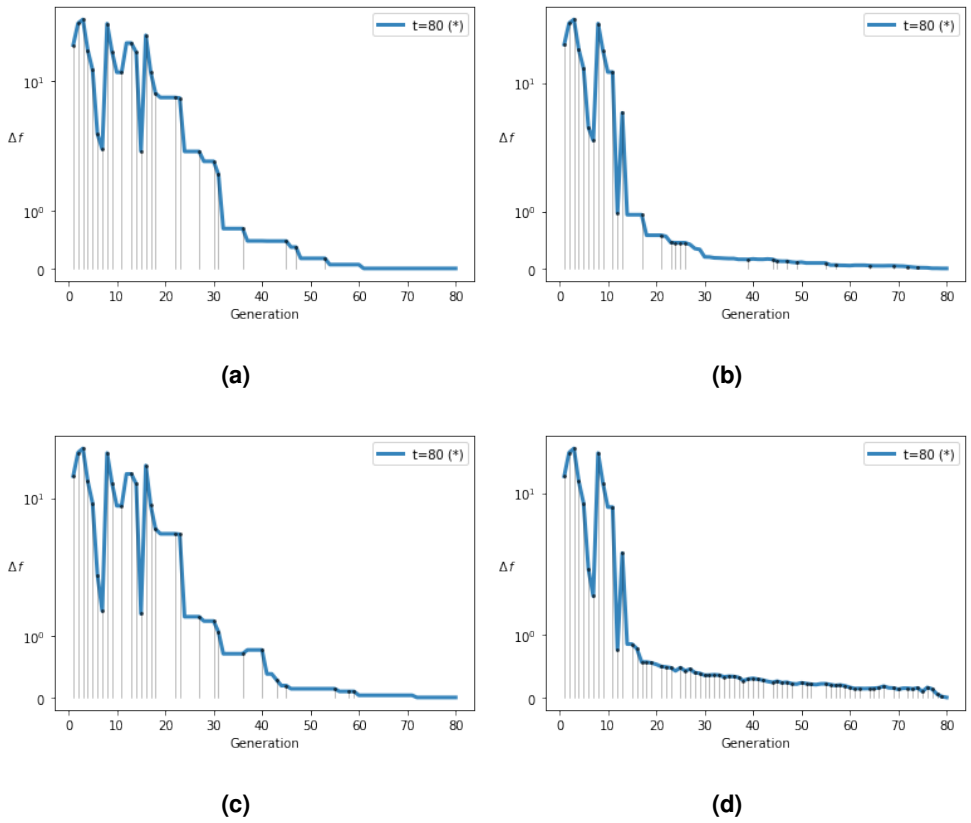


Figure B.1: Evolution of the objective space in 2019 for the BOUV (a), BOMV (b), MOUV (C), MOMV (d) pairs.