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

Miguel Carvalho Figueira
miguel.c.figueira@tecnico.ulisboa.pt

***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.**

***Index Terms*—Pairs Trading, Statistical Arbitrage, Machine Learning, Genetic Algorithms, Multi-objective Optimization**

I. Introduction

Pairs trading (PT) was introduced mid 80's and is one of the most popular arbitrage strategies. PT is characterized by being market neutral, as it attempts to avoid systematic risk and be profitable both in bullish and bearish markets. A further advantageous feature consists in focusing on the relative pricing of a security, bypassing the inconvenience of evaluating an asset.

The strategy is based on the idea that two securities that are close substitutes to each other, whether because they belong to same sector or are affected by the same economic factors, shall have similar prices (Law of One Price). These two assets constitute a pair. 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, by shorting the overvalued asset and buying the undervalued one.

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.

In fact, although PT performed well through the 80's and 90's, its profitability has been decreasing. [1] 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). [1] concludes that a naive method of pairing securities will not be enough to generate consistent returns.

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, should conciliate two opposite goals: minimize risk and maximize profit.

To address this dichotomy, in this work, pairs will be formed through a Multi-objective Optimization Problem (MOOP), and Genetic Algorithms (GA) will be used to form the best possible pairs. Furthermore, pairs composed of more than the standard two securities will be considered, 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 Machine Learning (ML) will be employed to forecast the spread of each pair and generate trading signals. In the forecasting-based trading module, an ARIMA and XGBoost models will be used to anticipate trading opportunities.

This work's objectives can be synthesized as in fig. 1. Research Stage (RS) I will address the formation of pairs, and RS II the trading strategy.

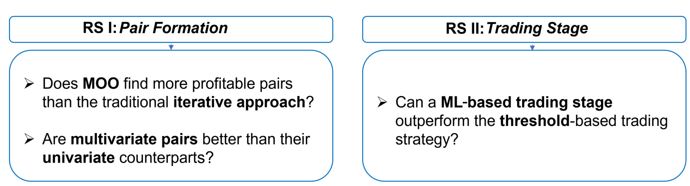

Fig. 1: Research questions to be answered.

The remainder of this manuscript is organized as follows: Section II provides a brief review of PT techniques found in the literature; Section III describes the pair formation methodology; Section IV delineates a forecasting-based strategy; Section V details the experiment design, and the features that ensure some proximity with a real trading environment;

Finally, Section VI presents the results obtained and Section VII summarizes the conclusions of this work.

II. Background and Related Work

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.

Following the comprehensive survey presented in [2], Pairs Trading approaches can be classified as:

- **PF:** Distance, Cointegration, Machine Learning.
- **TS:** Thresholds, Machine Learning.

A. Pair Formation

A standard pairing technique consists of grouping securities belonging to the same industry sector, with the intent of ensuring that they are influenced by the same external factors. This procedure can be observed in [1], [3]. On the other hand, an extensively iterative approach is used in [4], [5], 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 illustrates the various approaches.

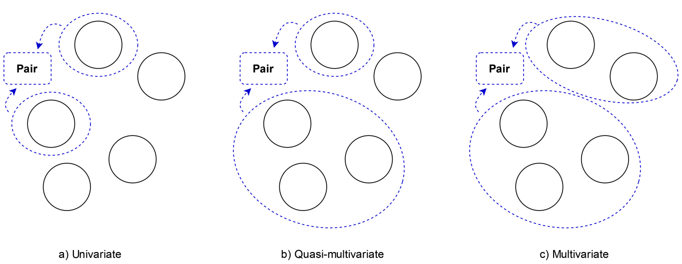

Fig. 2: Types of pairs regarding their cardinality.

The majority of the academic works on PT explores the univariate method, with rare exceptions, namely [6]–[8], which implement a quasi-multivariate strategy or [9], where a truly multivariate framework is tested. 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 or Cointegration criteria.

- Distance - the baseline method

Gatev et al. [3] use the Euclidean distance to evaluate pairs. The ones that present the smallest Sum of Squared Distances (SSD) between the two stocks during a rolling window of 12 months are selected.

An improvement to the SSD is suggested in [1]. 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.

- Cointegration

The cointegration approach tests the pair's spread for stationarity, namely with the Engle-Granger test. If two price series are cointegrated, their spread, given by

$$s_t = x_t - \beta y_t, \quad (1)$$

is stationary, where β is the cointegration factor. A trader can analyze a stationary spread to detect instants where it deviates from its habitual behavior, thus finding trading opportunities. According to [2], the key benefit of this strategy is the econometrically trustier association of identified pairs, opposing the distance method, which may find spurious relationships.

B. Trading Stage

Following the formation of the pairs, the Trading Stage encompasses deciding when to open and close a trade, by analyzing the pair's spread and making decisions according to its behavior. The typical procedure is to apply the same amount of capital to each component of the pair, i.e., the value obtained in the short is equal to the value spent on the long.

- Threshold-based strategy - standard approach

In the work of Gatev et. al. [3], 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 from the mean, the trade is opened. When the spread converges to its mean, the trade is closed. Other notable works already mentioned that follow this technique are [7], [9].

C. ML-based strategies

Table I encapsulates the ML techniques applied to a PT strategy. DQN refers to Deep Q-network, a Reinforcement Learning (RL) model.

TABLE I: Summary of ML techniques in PT.

Work	Methodology	Conclusions
[10]	(PF) OPTICS applied to the PF problem.	OPTICS achieved better returns than searching within a sector or iterating over all possible pairs.
[9]	(PF) NSGA II forms multivariate pairs.	Multi-objective formulation outperforms single objective. Multivariate PT presents low correlation with the market.
[4]	(TS) DDQN vs DQN as trading agents.	DDQN is more conservative and produces less volatile returns.
[5]	(TS) RL with DQN.	RL outperforms the standard method. Most profitable trades do not take very long to converge.
[6]	(TS) GA to optimize entry points.	Trading within a sector outperforms trading all types of stocks.

D. Benchmark Results

Table II compiles the returns obtained in several studies on the topic, that will be used to compare the ones attained is this work. The first three works use ML, while the last two employ standard techniques, as the Distance criterion and the Threshold-based strategy. Note that only [10] and [1] consider transaction costs.

TABLE II: Benchmark Results.

Work	Methodology	Data	Results
[4]	RL with DDQN	SP500 2018	131,3% ROI
[10]	OPTICS	Commodity ETFs 2015-2018	12,5% p.a.
[9]	NSGA II	SP500 2012-2016	1,72% monthly
[3]	Distance Threshold-based	SP500 1962-2002	11% p.a.
[1]	Distance with NZC	SP500's bank stocks 1962-2002	0,84% monthly

III. Proposed Pair Formation Framework

The PF module is designed to assess whether multivariate pairs formed with MOO perform better than the traditional pairs.

The motivation behind the multivariate technique relies on the idea that increasing the number of securities held reduces volatility, encapsulated in the financial mantra "Don't put all your eggs in one basket".

MOO will be applied to form pairs that simultaneously maximize profit and minimize risk.

A. Case Studies

Five different types of pairs, illustrated in fig. 3, will be formed to investigate the research goals above.

	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

Fig. 3: Case studies for different number of stocks and objective functions.

Serving as benchmark, 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], [3], 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 [9].

The objective functions for each type of pair are:

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

By minimizing the cointegration and maximizing the NZC, it is ensured that the pair exhibits a trustworthy relationship and strong mean-reversion, respectively. As such, the risk is minimized.

The half-life reflects how much time the pair takes to revert to its mean. By minimizing the half-life and maximizing the spread's volatility, the pair presents frequent and valuable trading opportunities, thus maximizing the profit potential.

B. Proposed Framework

Given the exponentially growing solution space induced by the combinatorial nature of multivariate pairs and the conflicting objectives that 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 obtain a diverse solution set. In particular, an elitist GA will be applied, NSGA II and III. Rudolph [11] demonstrates that GAs converge to the global optimal solution of some functions in the presence of elitism.

Some notable applications of NSGA II to portfolio management are found in [12], [13]. Moreover, [14] also uses NSGA III to select profitable stocks.

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

The adopted architecture of the PF module is delineated in fig. 4: 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. 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 RS I will be drawn.

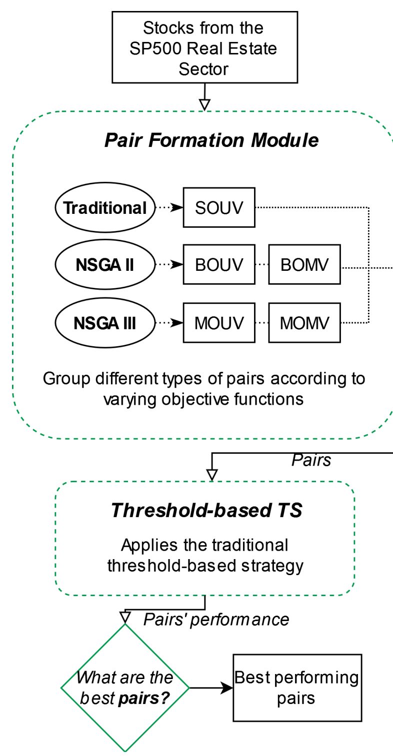

Fig. 4: Pair Formation module architecture.

C. 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 a 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:

$$
\begin{aligned}
\min_{c^1,c^2} \quad & f = [f_1, f_2, \ldots, f_M] \\
\text{s.t.} \quad & l \leq c^{1,2} \cdot \mathbf{1} \leq u && \text{(2a)} \\
& c_i^1 + c_i^2 \leq 1, \qquad i = 1, \ldots, N && \text{(2b)} \\
& coint(c^1, c^2) \leq 0.05 && \text{(2c)} \\
\text{and} \quad & c_j^{1,2} \in \{0, 1\}, \qquad j = 1, \ldots, N. && \text{(2d)}
\end{aligned}
$$

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

Constraint (2a) 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, $c^{1,2}$, by a vector of ones, $\mathbf{1}$. 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, (2b), states that stock j cannot belong to both components at the same time. Furthermore, the par must be cointegrated, hence constraint (2c). Finally, the component vectors will be composed of binary variables (2d).

- **GA representation**

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.

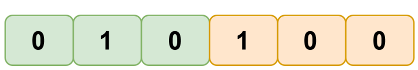

Fig. 5: Example chromosome for a universe of three stocks.

Fig. 5 illustrates a chromosome used to form a pair from a universe of three stocks, [A, B, C], thus $N = 3$. 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.

D. NSGA II and NSGA III

Since its introduction in [15], NSGA II is the reference when it comes to multi-objective optimization [16]. Its 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.
- 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.

However, with the increase of the number of objective functions, the number of non-dominated solutions increases exponentially. This leads to several negative consequences:

- New solutions have less representation in the population as the algorithm iterates.
- Diversity preservation becomes increasingly more complex.

NSGA III [17] is developed with these issues in mind. NSGA III is an evolution of the previous algorithm that is better suited to tackle many-objective optimization problems ($M \geq 3$). The algorithm differs from its precursor on how to filter the population and on the diversity-preserving mechanism.

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

- **Parameters**

Table III presents the parameters selected.

TABLE III: NSGA II and NSGA III parameters.

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

Both algorithms are applied using pymoo's implementation [18].

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 [19]. However, this method would require a much larger amount of time to converge. Furthermore, in the experiments performed, the evolution of the objective space stalls above the 80th generation.

NSGA III employs an energy-based method to generate reference directions, Riesz s-Energy, that is superior to the standard Das and Dennis's method since it allows the user to set a specific number of reference points [20].

E. Threshold-based Trading Strategy

The different types of pairs 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 IV displays the thresholds that dictate the strategy.

TABLE IV: 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 the 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. Most strategies normalize the spread as a *zcore*, simplifying the thresholds to -2, +2 and 0.

IV. Proposed Trading Framework

The traditional threshold-based trading model relies solely on the blindly defined thresholds that establish the entry and exit points of the strategy. Furthermore, the standard strategy only trades the signal in the direction of convergence, towards the mean. Consequently, 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.

The forecasting-based strategy proposed uses the day-ahead forecast of the spread to anticipate trading opportunities.

A. Trading Strategy

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

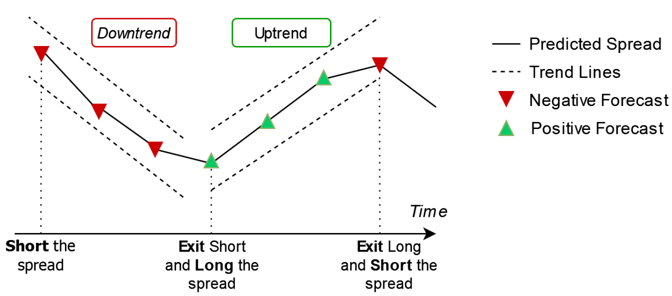

Fig. 6: Trend-based forecasting strategy.

Formally, the strategy tracks the predicted change in the spread's value, defined as $\Delta_{t+1} = \hat{s}_{t+1} - s_t$, and recommends a Long/Short position, $\hat{P}$, according to the following conditions, where TC are the transaction costs:

$$\hat{P} = \begin{cases} \text{if } \Delta_{t+1} > \text{TC}, \text{ Long position,} \\ \text{if } \Delta_{t+1} < -\text{TC}, \text{ Short position,} \\ \text{otherwise, remain outside the market.} \end{cases}$$

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. 6, 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.

B. Proposed Framework

The spread will be predicted with an ARIMA and XGBoost models. 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.

The ARIMA model was selected due to its simplicity and proven robustness in a wide field of forecasting applications [21]. 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.

XGBoost [22] provides a novel take on gradient boosting frameworks. It will be used by virtue of being highly flexible and capable of dealing with massive and complex data sets. Additionally, [23] concludes that frameworks that rely on trees produce superior results compared to ANNs such as RNNs, while being faster to train.

The structural design of the proposed TS is presented in fig.7. 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 trend. Finally, the performance of the trading system developed will allow to assess if a ML-based approach contributes positively to the trading stage.

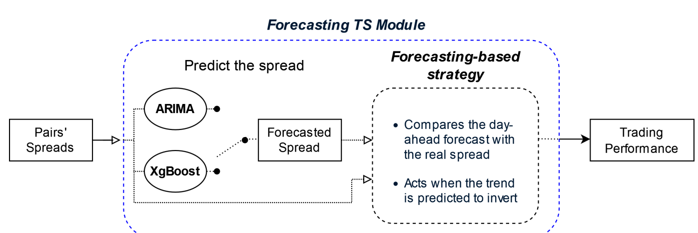

Fig. 7: Forecasting TS module architecture.

C. Models' parameters

- **ARIMA**

Three parameters that reflect and explain the underlying structure of the data 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 of the signals to forecast. The best configuration is presented in table V.

TABLE V: ARIMA and XGBoost parameters.

Model	Configuration
ARIMA	p = 5, d = 1, q = 0
XGBoost	n_estimators = 400, max_depth = 8 $\eta = 0.05$, $\lambda = 1$ $\gamma = 0.005$, patience = 5

Regarding p, it could be pointed that the spreads should be stationary, thus $p = 0$, since that is the ideal condition for a PT strategy that the pairs formed should satisfy. However, test spreads are rarely stationary, as it will be exposed in section VI-A.

- **XGBoost**

To take advantage of XGBoost's flexibility and adaptability, the model will be given the following features[1] (besides the past day's value): $\text{SMA}_{5,\ 10,\ 15}$, EMA_9, RSI, MACD.

Some regularization techniques were applied to the model, namely: restricting the max depth of each tree; the learning rate, η; and the pruning parameter, γ. Additionally, λ reduces the model's sensibility to an individual observation. Finally, an early stopping method was used. The best hyperparameters for this model were defined by applying a grid search and are displayed in table V.

V. Experiment Design

This section discusses some relevant aspects regarding the implementation, as the data set, the back-testing setup and the trading rules that try to mimic a real trading environment.

A. Data set - S&P 500 Real Estate sector

This work will use S&P 500 financial information. Stocks belonging to the Real Estate sector will be studied, 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.

Regarding Research Stage I (RS I), a 3-year formation period will be used, based on the assumption that selecting pairs across a relatively long time period reveals long-lasting associations. Then, the formation methods are tested in the next year. The sliding window is displayed in fig. 8.

The time frame selected for this work spans from 2018 to the end of 2021. The test years are highlighted in bold in fig. 8. The adopted period is considered relevant since it is recent - presenting 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.

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

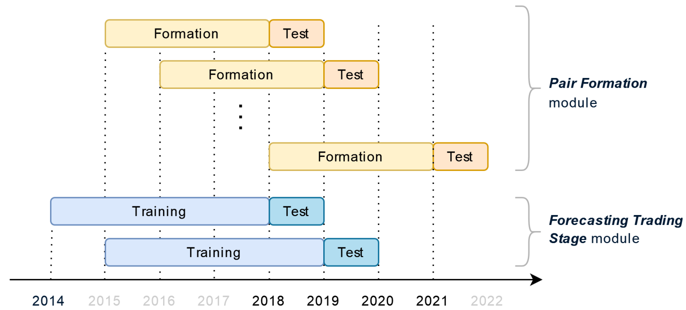

Fig. 8: Sliding window periods.

Concerning RS II, the Forecasting-based TS applies a 4-year training period followed by a 1-year test period. This process is executed for both algorithms, ARIMA and XGBoost, meaning a $(80\% - 20\%)$ train/test split. 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. The ARIMA model is fitted every day, and the XGBoost each 5 days.

B. Transaction Costs

The costs associated with each market operation silently erase the potential profit and should not be ignored. Table VI displays the fees considered, based on the estimations in [24].

TABLE VI: 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

C. Time series normalization and Look Ahead Bias

A crucial but frequently overlooked step in the traditional trading strategy is the spread normalization. The test spread, s, is normalized according to:

$$Z = \frac{s - \mu_s}{\sigma_s}, \tag{3}$$

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.

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 considered days. 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.

The best value for d, $d = 63$, or approximately three trading months, was obtained by iteratively testing different normalization periods.

D. Trading Portfolios

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.

An initial symbolic value of 1000$ is attributed to each portfolio. 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%, in the second year the portfolio will only have 950$ to start with. At the end of each trading year, every position is closed.

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. The amount applied to each component of the pair is the same and, at instant t, is described by:

$$p(t) = min(p_0, p(t-1)). \tag{4}$$

VI. Results

This section presents and analyzes the results obtained. Firstly, the pair formation techniques are examined in VI-A, and then the forecasting-based strategy is evaluated in VI-E.

A. Pair Formation Results

The number of pairs clustered for each case study is presented in table VII. It comes as no surprise that both

TABLE VII: Number of pairs selected for each formation method.

Formation Period	Cases				
	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

multivariate methods, BOMV and MOMV, select more pairs than their univariate equivalents, since that the combinatorial nature of multivariate pairs offers many more possible pairs.

Each of the two components of BOMV and MOMV pairs is composed of up to 5 stocks. The algorithms, through their natural selection filter, "opted for" having more stocks per component, 4 to 5, as it is possible to see in fig. 9.

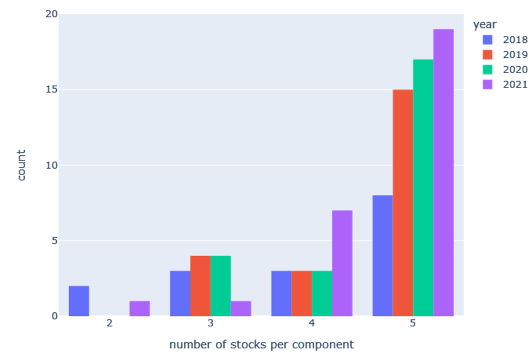

Fig. 9: Number of stocks per component for MOMV pairs.

TABLE VIII: Trading Performance of the different types of pairs.

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**
% non-convergent	66,67	63,89	72,13	76,00	69,67	69,93
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**
% non-convergent	75,00	75,00	33,33	50,00	58,33	56,25
BOMV						
ROI	2,71	-2,48	-14,31	-16,61	**-7,67**	**-28,43**
MDD	3,14	7,32	16,13	16,61	**10,78**	**32,50**
% non-convergent	85,19	17,65	95,65	65,79	66,07	69,52
MOUV						
ROI	0,42	-6,62	8,29	5,59	**2,01**	**7,59**
MDD	6,89	7,69	12,57	3,19	**7,58**	**17,95**
% non-convergent	85,71	100,00	70,00	55,56	77,82	75,76
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**
% non-convergent	50,00	100,00	75,00	42,86	66,69	66,67

B. Trading Performance

Table VIII summarizes the performance of the different PF methods.

A quick glance to the cumulative ROI[2] 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, revealing to be the superior technique and yielding $9,73\%$ yearly on average.

- Univariate vs. Multivariate pairs

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[3] are larger.

[2] Return On Investment.

[3] Maximum Drawdown: maximum observed loss from a peak to a trough of a portfolio.

This conclusion corroborates the finding of [9]: univariate pairs present themselves as more profitable.

- Forming pairs with MOO

Concerning the second research question: "Does MOO find 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.

C. Worst performing portfolios

The worst performing portfolios are the SOUV, BOMV, MOMV. Two major factors negatively influence these pairs: the pairs alter their behavior from the formation to the trading period; non-convergent pairs.

- Pairs change their behavior

To test whether the pairs still exhibit the desired proximity relationship by being cointegrated, table IX displays the number of pairs cointegrated in the testing period.

TABLE IX: Number of pairs still cointegrated in the testing period.

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

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 [25]. 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.

But how does the change in behavior negatively affect the returns? If the pair diverges long enough after a position is opened, the spread's mean, μ_s, will increase (recall that the mean is calculated with the past 3 months). Consequently, if the spread touches its mean, the pair is believed to have converged, and the trade is closed. However, that is far from the truth. The pairs' components are now more distant than ever, and the trade results in a loss, since it was bet that the components would meet again.

Fig. 10 illustrates the change in the behavior of the pair EQR-DLR, where the closely related price series suddenly distance themselves in the testing period.

This phenomenon is aggravated in multivariate pairs. Not a single BOMV or MOMV pair is cointegrated in the testing period. 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 explains the mediocre performance from the multivariate pairs, and further supports the conclusion that multivariate pairs are no better than univariate ones.

- Non-convergent pairs

Fig. 10: Different behavior of EQR-DLR in the training and testing periods.

An additional negative agent is the sizeable number of non-convergent pairs, i.e. pairs that, when the trading year ends, still present an open market position. 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? Fig. 11 displays the evolution of the BOUV portfolio in 2018, where the red dots signal the MDD. 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.

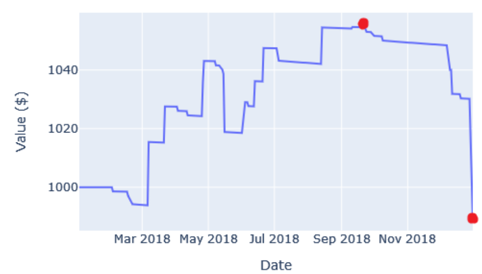

Fig. 11: Influence of non-convergence in BOUV pairs in 2018.

D. Best performing portfolio - BOUV

The BOUV pairs are the most profitable, and outperform the benchmark SOUV pairs, allowing to conclude that bi-objective optimization finds better pairs than the traditional iterative method based on cointegration.

Fig. 12 compares the BOUV portfolio with the SPY[4] index in 2020. The PT strategy outperforms the market, not only having a bigger ROI but also a lower MDD, signaled by the black dots. Given that 2020 was characterized by the March crash, consequence of the COVID-19 pandemic, this result confirms the idea found in the literature that PT is robust to market crashes.

[4]Benchmark that reflects the value of the S&P500 companies.

Fig. 12: BOUV and SPY in 2020.

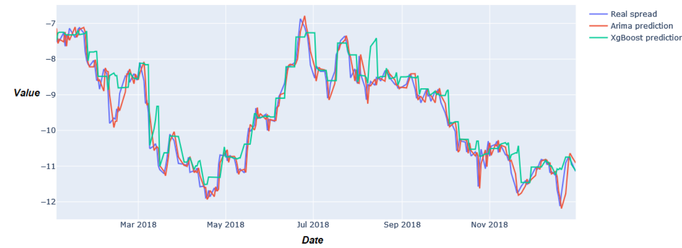

Fig. 13: Predicted KIM-ESS pair in 2018.

TABLE X: Forecasting performance of ARIMA and XGBoost.

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**

E. Forecasting Strategy Results

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

Table X summarizes the accuracy of the predictions of all pairs considered in 2018, 2019. The metric "percentage of spread's directions correctly predicted" reflects the number of times the model accurately forecasted if the spread's value would increase or decrease in the following day. The "A" or "X" in the model row stands for ARIMA or XGBoost.

The ARIMA model performs better than the XGBoost at predicting the spread's day-ahead value. The results are particularly curious given that the ARIMA model is much simpler in nature, and faster to train, suggesting that the extra complexity added by the XGBoost might be unnecessary. This conclusion is in accordance with [10], where an ARIMA model outperforms a LSTM.

Fig. 13 displays the predictions of both models for the KIM-ESS pair in 2018. While the ARIMA model follows the real spread 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.

On the other hand, the spread predicted by the XGBoost almost resembles a succession of step signals, since the XGBoost model is fitted each 5 days. Consequently, during the five-day 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. Hence, the XGBoost performs better when predicting the spread's direction.

F. Forecasting Trading Performance

Table XI 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.

TABLE XI: Forecasting-based strategy performance.

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**
XGBoost				
ROI	43,84	3,75	**23,80**	**49,24**
MDD	4,84	5,36	**5,10**	**5,36**
BOUV				
ROI	-1,11	-3,55	**-2,33**	**-4,62**
MDD	6,34	8,50	**7,42**	**8,50**

XGBoost obtains the best trading results, yielding $49,24\%$ of profit over the 2 years. Furthermore, it also outperforms the BOUV pairs. Additionally, the XGBoost portfolio has the lowest MDD, meaning that the portfolio does not decrease drastically in value at any given time. In fact, the biggest downtrend is only 5,36%, observed in 2019. Fig. 14 depicts the evolution of the portfolios compared with the SPY index. The MDD of the XGBoost portfolio is marked with black dots. The contrast in profitability from 2018 to 2019 is explained by analyzing the return for each trade. In 2019, the unsuccessful trades are much more damaging: a major negative action invalidates the gains obtained in a series of small, profitable trades.

The ARIMA portfolio results in drastic losses, explained by its bad performance in correctly predicting the future direction of the spread. To the ARIMA, the spread either goes up or

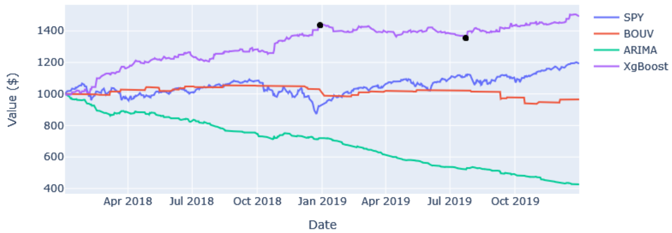

Fig. 14: Xgboost, ARIMA and BOUV portfolios compared with the SP500 in 2018-2019.

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 X. 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.

G. XGBoost - best overall portfolio

The XGBoost portfolio nets the best results, not only for the ML models but also including the portfolios studied in the PF module.

The performance of the XGBoost allows the conclusion that ML does improve a PT strategy. 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 of time to further validate this conclusion.

But how does the XGBoost portfolio fare against the benchmark results found in the literature? With an average return of 23,8%, the developed forecasting-based strategy outclasses [1], [3], that use non-ML techniques. In regards to ML-based strategies, this work outperforms the likes of [9], [10]. Nevertheless, the returns obtained fall short of the ones in [4], that achieves 33% yearly returns, with the caveat that it does not consider transaction costs.

VII. Conclusions

This work explored how ML techniques could improve a PT strategy. In the pair formation phase, multivariate pairs were created with MOO, using both NSGA II and III. Multivariate pairs were found to be inferior to univariate. Bi-objective optimization proved to be the best technique, yielding returns of 9,73% p.a. and beating the market during the Covid crash of 2020. Regarding the trading strategy, a novel trend-based approach obtained profits of 23% p.a., outperforming the returns of most of the works found in the literature. Additional contributions include finding that most pairs break their cointegration relationship in the trading period and emphasizing the negative influence of non-convergent pairs.

References

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

[2] 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

[3] 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.

[4] 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.

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

[6] 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.

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

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

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

[10] 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

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

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

[13] 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.

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

[15] 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.

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

[17] 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/

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

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

[20] 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.

[21] 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.

[22] 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

[23] 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.

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

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