α-dominance two-objective Optimization Genetic Programming for Algorithmic Trading under a Directional Changes Environment

Xinpeng Long, Michael Kampouridis
School of Computer Science and Electronic Engineering
University of Essex, Wivenhoe Park, United Kingdom
{xl19586, mkampo}@essex.ac.uk

***Abstract*— We present a novel genetic programming (GP) algorithm that combines physical time and event-based time indicators to trade on the stock market. Rather than only using data in fixed intervals (e.g. daily closing prices), we use directional changes to transform physical time into events and allow the GP to make trading decisions based on when significant price movements have occurred. We use a two-objective fitness function, which simultaneously optimizes return and risk. To overcome challenges with the convergence ability of the multi-objective GP, we apply an α-dominance strategy, which is able to relax the strict Pareto dominance criteria. We run experiments on 110 stocks from 10 international markets and compare results against a single-objective GP, as well as strategies based on technical analysis indicators and buy-and-hold. Results show that the proposed GP algorithm offers statistically significant improvements when compared to the above benchmarks.**

***Index Terms*—Directional Changes, Algorithmic Trading, Genetic Programming, Multi-Objective Optimization**

I. Introduction

Algorithmic trading refers to the process of executing financial transactions using algorithms that follow predefined rules. In recent years, applications of machine learning to algorithmic trading have gained popularity [1]. While algorithms usually view data from a physical-time perspective (e.g., daily closing prices, hourly data) [2], [3], event-based approaches have also been used. Directional changes (DC) is an event-based technique which summarises physical time into a series of events. These events are defined as the price movement over a user-defined threshold θ, such as a price change of 0.5%.

The first work that introduced the concept of directional changes (DC) in financial markets was [4] and was subsequently formalised by [5]. Since then, there's been a lot of works that have employed DC in trading, such as [6], [7], which developed trend following and contrarian DC strategies, and [8] which was applied to the foreign exchange market. There has also been several works that have combined machine learning with directional changes, e.g. [9], [10] applied classification and regression algorithms to forecast trend reversals, and [11] considered simultaneous recommendations from different DC thresholds under a genetic algorithm framework. In addition, [12] used a convolutional neural network long short-term memory (CNN-LSTM) model to predict DC events, and [13] demonstrated the advantages of combining DC indicators with technical analysis indicators. More recently, [14] used directional changes under a reinforcement learning environment.

While multi-objective optimisation has been used before for algorithmic trading, it has not been considered under a DC framework. The only exception to this is [15], which used the well-known NSGA-II algorithm [16] alongside genetic programming to optimise returns and risk. Using such multi-objective optimisation algorithms has the advantage of identifying trading strategies that consider multiple objectives simultaneously, rather than optimising a single aggregate objective, such as the Sharpe ratio, which is a common approach in the literature.

In this study, we present a novel two-objective (return, risk) optimisation GP that uses (i) features from DC and technical analysis, and (ii) α-dominance strategy [17] to compare solutions. While Pareto-based algorithms are very effective in optimising multiple objectives, they often cannot find a single solution that optimally satisfies all objectives simultaneously. With the introduction of α-dominance strategies, we are able to consider trade-offs between the conflicting objectives. We conduct experiments on 110 stocks from 10 different international markets and benchmark our results against a non-α-dominance multi-objective optimisation GP, as well as strategies from technical analysis and buy-and-hold. Our aim is to show that the proposed α-dominance multi-objective optimization DC algorithm is able to perform strongly in the stock market in terms of both return and risk.

The rest of this study is as follows. Section II presents relevant background information, while Section III presents our methodology. Section IV presents the experimental set up, and Section V the results of our experiments. Lastly, Section VI concludes this article and discusses future work.

II. Background

A. Overview of directional changes

Directional Changes form an event-based approach for summarising market price movements, as opposed to a fixed-interval-based approach. A DC event is identified only when the price moves towards one direction by a user-defined threshold. Depending on the direction of price movement, DC events could be categorized as upturn or downturn events. Frequently, after the confirmation of a DC event, an overshoot (OS) event follows; the OS event ends when a new price movement starts in an opposite trend, eventually leading to a new DC event. Algorithm 1 presents the pseudocode of DC.

Algorithm 1 Pseudocode for generating directional changes events.

Require: Initialise variables (event is Upturn event, $p^h = p^l = p(t_0), \Delta x_{dc}(Fixed) \geq 0, t_0^{dc} = t_1^{dc} = t_0^{os} = t_1^{os} = t_0$)

if event is Upturn Event **then**
  if $p(t) \leq p^h \times (1 - \Delta x_{dc})$ **then**
    $event \leftarrow DownturnEvent$
    $p^l \leftarrow p(t)$
    $t_1^{dc} \leftarrow t$ // End time for a Downturn Event
    $t_0^{os} \leftarrow t{+}1$ // Start time for a Downward Overshoot Event
  else
    if $p^h < p(t)$ **then**
      $p^h \leftarrow p(t)$
      $t_0^{dc} \leftarrow t$ // Start time for Downturn Event
      $t_1^{os} \leftarrow t-1$ // End time for an Upward Overshoot Event
    end if
  end if
else
  if $p(t) \geq p^l \times (1 + \Delta x_{dc})$ **then**
    $event \leftarrow UpturnEvent$
    $p^h \leftarrow p(t)$
    $t_1^{dc} \leftarrow t$ // End time for a Upturn Event
    $t_0^{os} \leftarrow t{+}1$ // Start time for an Upward Overshoot Event
  else
    if $p^l > p(t)$ **then**
      $p^l \leftarrow p(t)$
      $t_0^{dc} \leftarrow t$ // Start time for Upnturn Event
      $t_1^{os} \leftarrow t-1$ // End time for an Downward Overshoot Event
    end if
  end if
end if

The advantage of DC is that it offers traders a new perspective on price movements: focus on significant events and ignore other price movements that could be considered as noise. Directional changes have also led to the creation of DC-based indicators (similar to technical indicators), which allow traders to create novel trading strategies. We discuss some of these indicators in Section III.

B. NSGA-II

NSGA-II (Non-dominated Sorting Genetic Algorithm II) stands as an advanced genetic algorithm for multi-objective optimization [16], and is the algorithm we use in this paper. Its non-dominated sorting approach, which groups solutions into distinct fronts based on their dominance relationships, ensures a diverse range of solutions. Consequently, within each front, no single solution prevails over others. Dominance, according to the Pareto dominance principle, occurs when one solution outperforms another in at least one objective without deteriorating in any other. Given multiple fronts, individuals in lower-ranked fronts dominate those in higher-ranked ones (e.g., individuals in front 1 dominate those in front 2). To uphold diversity within each front, NSGA-II uses a crowding distance, which promotes solution dispersion by considering the solution density around each candidate.

During selection, NSGA-II initially prioritizes individuals based on their Pareto front rank, favoring those with lower ranks. In cases where individuals share the same Pareto front rank, preference is given to those with higher crowding distance. Subsequently, after the formation of a new population via genetic operators, both the original parent and new population undergo evaluation and ranking.

C. α-dominance strategy

While NSGA-II has been very successful, its strict Pareto dominance criterion can lead to some solutions dominating others even when they exhibit extreme superiority in one objective, but perform poorly in others—this was something we also observed in previous experiments [15]. To overcome this, in this paper we use an α-dominance strategy [17]. This relaxes the strict dominance criteria, by introducing a parameter α that controls the trade-offs among objectives. This strategy allows for weak trade-offs between objectives, providing a more flexible way to compare solutions. Equation 1 presents the α-dominance strategy:

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

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

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

Choosing an appropriate α_{ij} is crucial and three adaptation schemes have been proposed [17] to determine suitable values. These adaptation schemes include linear, sigmoid, and cosine functions, each defined as follows:

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

where C is a sufficiently large constant.

In our work, we will use each one of these adaptation schemes. We will further discuss this in our methodology section (Section III).

III. Methodology

We employ a genetic programming (GP) algorithm to create trading strategies. The GP's terminals use indicators from directional changes, as well as technical analysis. The GP simultaneously optimises returns and risk by using NSGA-II with α-dominance.

A. Representation

The Function set consists of the following logic operators: AND, OR, >, <. The Terminal set consists of 28 DC indicators (listed in Table I), 28 physical time technical analysis indicators (listed in Table II), and an ephemeral random constant (ERC), which takes values between 0 and 1. All indicators are normalized in this range. Figure 1 shows a sample tree that can be produced by the GP. Note that only Part 1 is evolved by the GP, while Part 2 remains the same throughout the evolutionary process. The root note is always an If-Then-Else (ITE) function. The first branch checks if the physical analysis OSV indicator is greater than 0.22 and if the NDC_10 DC indicator (total number of DC events of the last 10 days) is greater than -0.68. If both statements are true, then the tree returns a true statement, and the second branch of the ITE statement is invoked. In that case, a signal of 1 is generated, which denotes a 'buy' action, and hence one amount of stock is bought. If the first branch of the ITE returns false, then the third branch is invoked, and a 'hold' (0) action takes place.

Sell actions are not part of the GP tree. To decide when a 'sell' action will occur, we use the GP tree's output (buy or hold signal) to answer the following question: "Is the stock price going to increase by r% within the next n days?". To sell a previously bought stock we wait until one of the following two cases occur: (i) the stock price increases by r% within the next n days, or (ii) n days have passed since our initial purchase of the stock. Note that short-selling is not allowed in this trading strategy. Whenever a trade is completed (a buy action and a sell action are achieved), we calculate and record the buy and sell prices, P_b and P_s, respectively. All buy and sell actions factor in a 0.025% transaction cost.

B. Model evaluation

Following the initialization of the population, the next step involves the assessment of each individual. To evaluate each individual's performance, we consider two different financial metrics: rate of return (ROR), and risk. The former is a maximisation objective, and the latter a minimisation. Equations 3 - 4 present the three metrics.

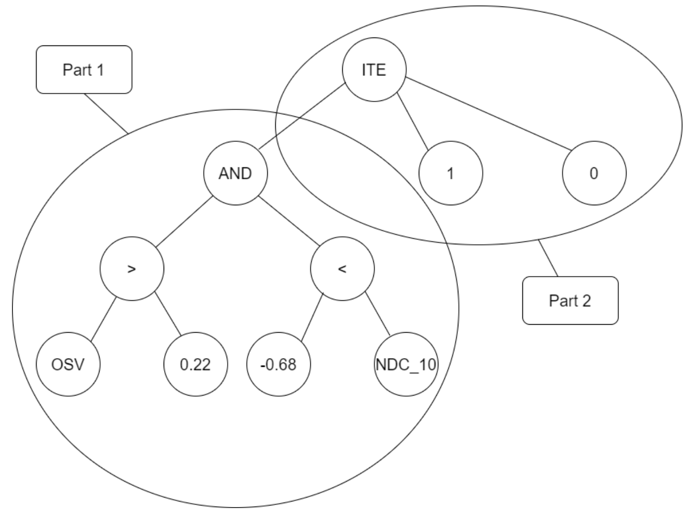


Fig. 1: Sample GP tree. If OSV is greater than 0.22 and N_{DC} for 10 days is greater than -0.68, then we get a signal for a buy action; otherwise, we hold; 0.22 and -0.68 are two random values generated by ERC.

$$\text{ROR} = \frac{(1-c)\cdot P_s - (1+c)\cdot P_b}{(1+c)\cdot P_b} \cdot 100\% \tag{3}$$

$$\text{Risk} = \sqrt{\text{Var(RoR)}} \tag{4}$$

where P_s refers to the sell price, P_b refers to the buy price, and c is the transaction cost.

For completeness, we also introduce in Equation 5 an aggregate metric, namely the Sharpe ratio. We will use the Sharpe ratio in our experiments as the fitness function of a single-objective GP algorithm, which will act as a benchmark to our multi-objective GPs.

$$\text{Sharpe ratio} = \frac{\text{E(RoR)} - R_f}{\text{Risk}}, \tag{5}$$

where E(RoR) stands for the sample mean of the list of the rate of returns for a given stock and R_f is the risk-free rate, which in our experiments is equal to 0.022%.

C. Selection and Genetic operators

We use tournament as the GP selection method. To select the winner of a tournament, NSGA-II considers the Pareto front rank and the crowding distance. Hence, after obtaining a k number of individuals for the tournament, we first compare the Pareto front rank—the individual with a lower rank survives. For individuals with equal Pareto front rank, the one with a higher crowding distance is selected.

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

TABLE I: DC indicators; see also [18]

Indicator	Description	Periods (days)
TMV	TMV is the price movement between the extreme point at the beginning and end of a trend, normalized by the threshold θ.	N/A
OSV	OSV is defined as the percentage difference between the current price and the last directional changes confirmation price divided by the threshold θ.	N/A
Average OSV	This is the average value of OSV over the selected period.	3, 5, 10
R_{DC}	R_{DC} represents the time-adjusted return of DC. It could be calculated as TMV times threshold θ divided by the time intervals between each extreme point.	N/A
Average R_{DC}	This is the average value of R_{DC} over the selected period.	3, 5, 10
T_{DC}	This is the time spent on a trend.	N/A
Average T_{DC}	This is the average value of T_{DC} over the selected period.	3, 5, 10
N_{DC}	N_{DC} is the total number of DC events over the selected period.	10, 20, 30, 40, 50
C_{DC}	C_{DC} is defined as the sum of the absolute value of TMV over the selected period.	10, 20, 30, 40, 50
A_T	A_T represents the difference between the time DC spends on the up trends and down trends over the selected period.	10, 20, 30, 40, 50

TABLE II: Physical time (technical analysis) indicators; see also [19]

Indicator	Description	Periods (days)
MA	Moving average for a given period.	10, 20, 30, 40, 50
CCI	Commodity Channel Index, which measures the deviation of an asset's price from its statistical average.	10, 20, 30, 40, 50
RSI	Relative Strength Index, which is a momentum oscillator to measure the magnitude of recent price changes and determine overbought or oversold conditions of an asset.	10, 20, 30, 40, 50
William's %R	Measures oversold or overbought conditions of an asset by comparing the closing price of an asset to its price range over a set period of time.	10, 20, 30, 40, 50
ATR	Average True Range, which measures the volatility of an asset by calculating the average of the true range over a set period of time.	3, 5, 10
EMA	Exponential Moving Average, which calculates a weighted average of a series of prices over a set period of time, where more recent prices are given greater weight in the calculation.	3, 5, 10
OBV	On Balance Volume, which measures buying and selling pressure, by calculating the cumulative total of an asset's volume, where positive volume is added to the total of an up day and negative volume is subtracted on a down day.	N/A
PSAR	Parabolic Stop and Reverse, which identifies potential reversals in the direction of an asset's price movement by placing dots on a chart that indicate potential stop and reverse points for a long or short position.	N/A

D. α-dominance strategy

As previously mentioned, to overcome problems with Pareto's strict dominance criterion, we use the α-dominance strategy, which was introduced in Equation 1. The proposed multi-objective optimization GP algorithm has two objectives: rate of return (ROR) and risk. Hence, the objectives i and j from the original Equation 1 now become ROR and risk, respectively; in addition, because the two objectives are conflicting (one maximization, one minimization), the sign in front of the α parameter from Equation 1 needs to change from positive (+) to negative (-). Hence, the α-dominance strategy for rate of return is:

$$G_{ROR}(A,B) = f_{ROR}(A) - f_{ROR}(B) - \alpha_{ROR,Risk}(f_{Risk}(A) - f_{Risk}(B)) \quad (6)$$

Similarly, the α-dominance strategy for risk is:

$$G_{Risk}(A,B) = f_{Risk}(A) - f_{Risk}(B) - \alpha_{Risk,ROR}(f_{ROR}(A) - f_{ROR}(B)) \quad (7)$$

Furthermore, as mentioned in Section II-C, there are three adaptation schemes to determine the value of $\alpha_{i,j}$, namely $f_{linear}, f_{sigmoid}$, and f_{cosine} (see Equation 2). In our experiments, we apply each one of these three adaptation schemes to Equations 6 and 7. Hence, we end up with six different α-dominance strategy configurations: (i) ROR with sigmoid adaptation ($\alpha MOO2^{Sig}_{ROR}$), (ii) ROR with cosine adaptation ($\alpha MOO2^{Cos}_{ROR}$), (iii) ROR with linear adaptation ($\alpha MOO2^{Lin}_{ROR}$), (iv) Risk with sigmoid adaptation ($\alpha MOO2^{Sig}_{Risk}$), (v) Risk with cosine adaptation ($\alpha MOO2^{Cos}_{Risk}$), and (vi) Risk with linear adaptation ($\alpha MOO2^{Lin}_{Risk}$). 'MOO2' refers to the fact that each algorithm is a two-objective optimization algorithm.

IV. Experimental set up

A. Data

The data used in this experiment comprises daily price data from 10 arbitrarily selected stocks for each of 10 international markets, namely DJIA, NASDAQ, NYSE, the Russell 2000 Index, and S&P500 in the United States, NIFTY 50 in India, TSEC in China (Taiwan), DAX index in Germany, Nikkei 225 in Japan, and FTSE 100 index in the UK. Data was selected for the 10-year period from 25/11/2010-24/11/2020. This resulted in a total of 110 datasets (10 markets × 10 stocks + 10 indices). The data was sourced from Yahoo! Finance. Our evaluation methodology involved dividing each dataset into three separate parts: the initial 60% of the data being the training set, the subsequent 20% being the validation set and the final 20% being the testing set. During parameter tuning, algorithms are trained using the training set and evaluated on the validation set. Once the optimal set of parameters has been determined, the GP is trained one final time in the combined training and validation set, and subsequently applied to the test set to assess the performance of the model. More information about tuning is provided in Section IV-C.

B. Benchmarks

As the proposed $\alpha MOO2$ algorithms are using the α-dominance strategy, we benchmark them against a MOO2 without this dominance strategy (i.e. a GP algorithm using NSGA-II to optimize rate of return and risk). We also benchmark them against a single-objective optimization (SOO) GP-based approach that employs the Sharpe ratio as its fitness function.

Furthermore, we benchmark the $\alpha MOO2$ algorithms against trading strategies derived from three popular technical analysis indicators, as representatives of a physical time setup: moving average convergence divergence (MACD), on balance volume (OBV), and momentum (MTM). For MACD and OBV, a buy signal is produced when their long-term moving average (50 days) has a higher value than their short-term moving average (10 days); and vice versa for a sell action. For MTM, a buy signal is generated when the current price is higher than the price 10 days ago, and vice versa for the sell action. All of the above values (10 and 50 days for MACD and OBV; 10 days for MTM) were selected through grid search.

Lastly, we also compare the proposed $\alpha MOO2$ algorithms against the passive trading strategy of buy-and-hold, which is a popular benchmark against active trading strategies, such as the GPs utilized in this paper.

C. Parameter tuning

We performed a grid search to decide on the optimal GP parameters, and tuning took place on the validation set. Table III shows the parameters' values after tuning.

TABLE III: Parameters of the GP algorithm

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

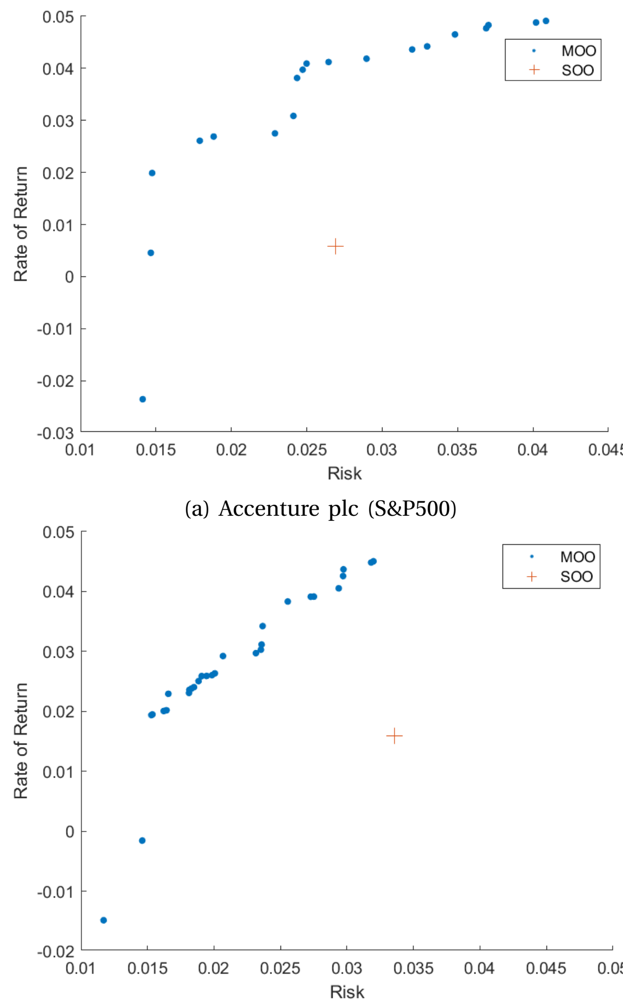
Regarding the trading strategy, recall that there are three parameters, two parameters derived for the question "whether the stock price will increase by $r\%$ during the next n days ?" and one parameter is the threshold on DC. Rather than tuning the above parameters and then selecting the best set across all datasets (which is what we did for the GP), we decided to allow for tailored values for each dataset.

V. Result and Analysis

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

A. Pareto front

Figure 2 shows the Pareto front for the $\alpha MOO2^{cos}_{ROR}$ algorithm, along with the best model (highest Sharpe ratio) from the SOO GP, which was selected across the 50 independent GP runs. The $\alpha MOO2^{cos}_{ROR}$ algorithm was selected as a representative example - results are similar for the other multi-objective GPs. As we can observe from the two plots, there are several solutions on the Pareto front. For Accenture (Figure 2a), the majority of solutions have a higher rate of return than the one by SOO. There are also several solutions that dominate the SOO's model, as they have both higher return and lower risk. For Experian (Figure 2b) the results are even better for $\alpha MOO2^{cos}_{ROR}$, as the majority of its solutions dominate the SOO solution.



(a) Accenture plc (S&P500)

(b) Experian plc (FTSE100)

Fig. 2: Comparison of the Pareto front solutions with the best single objective optimization GP.

B. Best model results

Although multi-objective optimization algorithms are able to find solutions in the Pareto front that dominate solutions from the SOO GP, in the real world a trader would be interested in identifying a *single trading strategy* to use in the market. Hence, this section provides a comparison between the best model from SOO and the 'best' model from the MOO2 algorithms. To select the best model from SOO we look into the 50 independent runs in training, find the best model (in terms of Sharpe ratio) and report its test set value. With regards to the MOO2 algorithms, although we cannot strictly talk about a 'best' solution in the Pareto front, we needed to define a way of selecting a single solution to compare it with the SOO. Given that the two objectives of MOO2 are rate of return and risk, which are also the components of SOO's fitness function of Sharpe ratio, we decided to look into the 50 independent GP runs' *Pareto fronts* from the final generation in the training set and select the model with the best Sharpe ratio; we then report the model's Sharpe ratio in the test set.

Table IV presents the performance of the SOO, MOO2, and $\alpha MOO2$ algorithms. We can observe that $\alpha MOO2^{Cos}_{ROR}$ achieves the best average (1.84%) and median (1.68%) value for rate of return, while SOO observes the worst average (0.98%) and median (0.82%) values. Regarding risk (middle part of Table IV), SOO is the top performer, followed by the $\alpha MOO2$ algorithms, which exhibit similar risk levels; MOO2 has slightly worse risk figures across all statistics. Lastly, the bottom part of the table presents the Sharpe ratio of the portfolio of 110 stocks (assuming equal weight) per algorithm. Note that this is a single value for the portfolio, hence no summary statistics are presented for Sharpe ratio. As we can observe, $\alpha MOO2^{Sig}_{ROR}$ has the highest value (0.76). This figure is double that of the benchmark SOO algorithm.

We also conduct pairwise comparisons between SOO and the MOO2 algorithms using the Kolmogorov-Smirnov (KS) non-parametric statistical test to assess the significance of the observed results. Our focus was on understanding the enhancements brought about by each MOO2 algorithm over the SOO. The null hypothesis is that the two distributions originate from the same continuous distribution. Given the seven pairwise comparisons, we apply the Holm-Bonferroni correction to account for these multiple comparisons. Consequently, the minimum acceptable p-value for achieving statistical significance at a 5% level is determined by α(rank) = 0.05, where rank $\in$ {1, 2, 3, 4, 5, 6, 7}, and 7-rank+1, which varies for different ranks of the p-values found. The rank denotes the order of magnitude of the p-values, with 1 representing the smallest and 7 the largest. The ranked p-values reveal significant differences between the samples: the first p-value should be less than 0.0071, the second less than 0.0083, the third less than 0.01, the fourth less than 0.0125, the fifth less than 0.0166, the sixth less than 0.025, and the seventh less than 0.05.

Table V presents the KS test results for rate of return (Va) and risk (Vb). As observed, all multi-objective optimization algorithms are statistically and significantly outperforming SOO's rate of return. In terms of risk, MOO2's poor performance is statistically outperformed by SOO. No other statistical differences among the αMOO2 algorithms and SOO are observed, which confirms our earlier observation that the $\alpha MOO2$ algorithms have similar risk performance with SOO. We can thus conclude that the $\alpha MOO2$ algorithms have statistically and significantly improved the rate of return over the SOO algorithm, while they maintained similar risk levels.

C. Comparison of $\alpha MOO2$ algorithms to technical analysis trading strategies

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

Table VI compares the performance of the two MOO2 algorithms and the technical indicators. Both $\alpha MOO2^{Sig}_{ROR}$ and $\alpha MOO2^{Cos}_{ROR}$ have high average and median rate of return values. MACD appears to be competitive when looking at its average value (1.71%), but this is only because of outliers; its median value is only 0.14%. The two $\alpha MOO2$ algorithms also have the lowest risk values for both average (0.07) and median (0.06), which are almost 50% lower than the risk values of MACD, OBV, and MTM. Lastly, $\alpha MOO2^{Sig}_{ROR}$ has the highest Sharpe ratio value (0.76), closely followed by $\alpha MOO2^{Cos}_{ROR}$, while the highest value by a technical indicator is 0.28 (MTM).

In order to compare the performance among the different algorithms (rather than doing pairwise comparisons as earlier)[1], we run the Friedman non-parametric test, where we calculate the average rank of each algorithm in terms of return and risk—the lower the rank, the better the algorithm's performance. We also perform the Bonferroni post-hoc test, and present both in Table VII. For each algorithm, the table shows the average rank (first column), and the adjusted p-value of the statitsical test when that algorithm's average rank is compared to the average rank of the algorithm with the best rank (control algorithm) according to Bonferroni's post-hoc test (second column) [20], [21]. As we can observe, for both return and risk $\alpha MOO2^{Cos}_{ROR}$ ranks first and statistically outperforms all technical indicators' strategies.

D. Buy and hold

We now compare the two $\alpha MOO2$ algorithms with the buy-and-hold strategy. A critical aspect to note is that the

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

TABLE IV: Comparison between SOO and MOO2 algorithms. Best values per row appear in boldface.

Measurement				Rate of Return				
Algorithm	SOO	MOO2	$\alpha MOO2^{Sig}_{Risk}$	$\alpha MOO2^{Sig}_{ROR}$	$\alpha MOO2^{Cos}_{Risk}$	$\alpha MOO2^{Cos}_{ROR}$	$\alpha MOO2^{Lin}_{Risk}$	$\alpha MOO2^{Lin}_{ROR}$
Average	0.98%	1.64%	1.65%	1.67%	1.81%	**1.84**%	1.49%	1.65%
Median	0.82%	1.55%	1.66%	1.66%	1.55%	**1.68**%	1.46%	1.50%
Standard deviation	**0.01**	0.04	0.02	0.01	0.02	0.02	0.02	0.02
Max	6.00%	**29.91**%	8.22%	7.98%	11.35%	7.21%	8.28%	9.24%
Min	-3.12%	-6.04%	-5.22%	-**1.91**%	-2.34%	-2.18%	-2.54%	-4.45%
Measurement				Risk				
Algorithm	SOO	MOO2	$\alpha MOO2^{Sig}_{Risk}$	$\alpha MOO2^{Sig}_{ROR}$	$\alpha MOO2^{Cos}_{Risk}$	$\alpha MOO2^{Cos}_{ROR}$	$\alpha MOO2^{Lin}_{Risk}$	$\alpha MOO2^{Lin}_{ROR}$
Average	**0.06**	0.09	0.07	0.07	0.07	0.07	0.07	0.07
Median	**0.05**	0.07	0.06	0.06	0.06	0.06	0.06	0.06
Standard deviation	**0.04**	0.06	0.05	0.05	**0.04**	0.05	**0.04**	0.06
Max	0.32	0.45	0.32	0.33	0.28	0.29	**0.24**	0.46
Min	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Measurement				Sharpe ratio				
Algorithm	SOO	MOO2	$\alpha MOO2^{Sig}_{Risk}$	$\alpha MOO2^{Sig}_{ROR}$	$\alpha MOO2^{Cos}_{Risk}$	$\alpha MOO2^{Cos}_{ROR}$	$\alpha MOO2^{Lin}_{Risk}$	$\alpha MOO2^{Lin}_{ROR}$
	0.38	0.40	**0.76**	0.61	0.71	0.70	0.71	0.59

TABLE V: Kolmogorov-Smirnov tests between SOO (control) and MOO2 algorithms (first column) for ROR and risk. *p*-values (second column) below the adjusted significance level (third column) appear in boldface to indicate statistical significance at 5% level. The calculation of the adjusted significance level is shown in brackets in the third column.

(a) Rate of Return

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

(b) Risk

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

buy-and-hold strategy involves a single transaction over the entire period under study, where we buy one unit of stock on the first day of trading and sell it on the last day; hence, the risk metric cannot be calculated. Furthermore, due to buy-and-hold making only a single trade (while the MOO2 GPs make several), it is fairer to compare them across the total return over the test set period rather than in terms of rate of return.

Table X presents the performance metrics of the MOO2 algorithms alongside the buy-and-hold strategy. Buy-and-hold has the highest average value, but this is only because of outliers. When looking at the median, the two MOO2 algorithms have around 7% higher total return. We again use the non-parametric Friedman test to support the analysis. $\alpha MOO2^{Cos}_{ROR}$ ranks first and statistically and significantly outperformed buy-and-hold, as it can be see from Table XI.

VI. Conclusion

To conclude, this paper presented a novel multi-objective genetic programming algorithm, which incorporates features from the event-based concept of directional changes. Our results showed that the proposed α-dominance algorithms are able to statistically and significantly outperform the single-objective GP in terms of rate of return, without compromising on risk. In addition, they were able to outperform trading strategies derived from technical analysis indicators, as well as the common passive investment benchmark of buy-and-hold. Future work will focus on exploring different multi-objective setups, e.g. considering more than the two objectives presented in this paper. We will also explore different multi-objective optimization algorithms.

References

[1] A. Brabazon, M. Kampouridis, and M. O'Neill, "Applications of genetic programming to finance and economics: past, present, future," *Genetic Programming and Evolvable Machines*, vol. 21, pp. 33–53, 2020.

[2] X. Long, M. Kampouridis, and D. Jarchi, "An in-depth investigation of genetic programming and nine other machine learning algorithms in a financial forecasting problem," in *2022 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2022, pp. 01–08.

[3] E. Christodoulaki, M. Kampouridis, and M. Kyropoulou, "Enhanced strongly typed genetic programming for algorithmic trading," in *Genetic and Evolutionary Computation Conference (GECCO)*. ACM, 2023.

[4] D. M. Guillaume, M. M. Dacorogna, R. R. Davé, U. A. Müller, R. B. Olsen, and O. V. Pictet, "From the bird's eye to the microscope: A survey of new stylized facts of the intra-daily foreign exchange markets," *Finance and stochastics*, vol. 1, no. 2, pp. 95–129, 1997.

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

Rate of return					
	$\alpha MOO2^{Sig}_{ROR}$	$\alpha MOO2^{Cos}_{ROR}$	MACD	OBV	MTM
Average	1.65%	**1.84**%	1.71%	0.77%	0.32%
Median	1.66%	**1.68**%	0.14%	0.18%	0.11%
StDev	0.02	0.02	0.06	0.05	**0.01**
Max	8.22%	7.21%	36.01%	**35.65**%	6.95%
Min	-5.22%	-2.18%	-9.04%	-11.11%	**-1.48**%
Risk					
	$\alpha MOO2^{Sig}_{ROR}$	$\alpha MOO2^{Cos}_{ROR}$	MACD	OBV	MTM
Average	**0.07**	**0.07**	0.12	0.12	0.15
Median	**0.06**	**0.06**	0.07	0.09	0.09
StDev	**0.05**	**0.05**	0.22	0.12	0.20
Max	0.32	**0.29**	1.58	0.76	1.82
Min	0.01	0.01	**0.00**	**0.00**	0.03
Sharpe ratio					
	$\alpha MOO2^{Sig}_{ROR}$	$\alpha MOO2^{Cos}_{ROR}$	MACD	OBV	MTM
	0.76	0.70	0.24	0.06	0.28

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

TABLE VIII: Rate of Return

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

TABLE IX: Risk

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

TABLE X: Summary statistics for the best two MOO2 algorithms and buy-and-hold in terms of total return. Best value per row appears in boldface.

Algorithm	$\alpha MOO2^{Sig}_{ROR}$	$\alpha MOO2^{Cos}_{ROR}$	Buy-and-hold
Average	30.53%	32.50%	**41.11%**
Median	**19.71%**	19.02%	11.44%
Standard deviation	**0.40**	0.46	1.81
Max	249.47%	294.70%	**1753.05%**
Min	**-23.00%**	-31.20%	-89.62%

TABLE XI: Friedman test with Bonferroni's post-hoc test between $\alpha MOO2$ and buy-and-hold. Statistically significant differences at 5% level shown in boldface.

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

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

[6] M. Aloud, "Modelling the fx market traders' behaviour: an agent-based approach, chapter 15, alexandrova-kabadjova b., s. martinez-jaramillo, al garcia-almanza & e. tsang (ed.), simulation in computational finance and economics: Tools and emerging applications," 2012.

[7] ——, "Directional-change event trading strategy: Profit-maximizing learning strategy," in *Proceedings of the Seventh International Conference on Advanced Cognitive Technologies and Applications, Nice, France*, vol. 22, 2015.

[8] A. Bakhach, E. Tsang, W. L. Ng, and V. R. Chinthalapati, "Backlash agent: A trading strategy based on directional change," in *2016 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 2016, pp. 1–9.

[9] A. Adegboye and M. Kampouridis, "Machine learning classification and regression models for predicting directional changes trend reversal in fx markets," *Expert Systems with Applications*, vol. 173, p. 114645, 2021.

[10] A. Adegboye, M. Kampouridis, and F. Otero, "Improving trend reversal estimation in forex markets under a directional changes paradigm with classification algorithms," *International Journal of Intelligent Systems*, vol. 36, no. 12, pp. 7609–7640, 2021.

[11] ——, "Algorithmic trading with directional changes," *Artificial Intelligence Review*, vol. 56, no. 6, pp. 5619–5644, 2023.

[12] A. Rostamian and J. G. O'Hara, "Event prediction within directional change framework using a cnn-lstm model," *Neural Computing and Applications*, vol. 34, no. 20, pp. 17 193–17 205, 2022.

[13] X. Long, M. Kampouridis, and P. Kanellopoulos, "Genetic programming for combining directional changes indicators in international stock markets," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2022, pp. 33–47.

[14] G. Rayment and M. Kampouridis, "High frequency trading with deep reinforcement learning agents under a directional changes sampling framework," in *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2023, pp. 387–394.

[15] X. Long, M. Kampouridis, and P. Kanellopoulos, "Multi-objective optimisation and genetic programming for trading by combining directional changes and technical indicators," *IEEE XPlore*, 2023.

[16] 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, no. 2, pp. 182–197, 2002.

[17] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of pareto-based moeas: Does non-dominated really mean near to optimal?" in *Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546)*, vol. 2. IEEE, 2001, pp. 957–962.

[18] M. E. Aloud, "Time series analysis indicators under directional changes: The case of saudi stock market," *International Journal of Economics and Financial Issues*, vol. 6, no. 1, pp. 55–64, 2016.

[19] A. Kelotra and P. Pandey, "Stock market prediction using optimized deep-convlstm model," *Big Data*, vol. 8, no. 1, pp. 5–24, 2020.

[20] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, p. 1 – 30, 2006.

[21] S. Garcia and F. Herrera, "An extension on" statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons." *Journal of machine learning research*, vol. 9, no. 12, 2008.