Multi-objective optimisation and genetic programming for trading by combining directional changes and technical indicators

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

***Abstract*—Directional changes (DC) have been shown to form an effective approach in algorithmic trading by converting fixed time series into event-based series and focusing on key events. Previous work has focused on forecasting the inflection point in the market and proposing new indicators under the DC framework, with just a handful of papers concerned with training and using DC indicators through machine learning. Earlier research has shown that genetic programming (GP) combining DC and physical time indicators could achieve positive returns with low risk. However, the fitness function used in that work is simply a risk-adjusted return. In this paper, we investigate whether a multi-objective optimisation approach could improve the performance of GP-based strategies in the market. We evaluate the cumulative return, risk, and rate of return of the proposed approach under 110 datasets from 10 different markets. Furthermore, we compare the proposed strategy against GP-based single objective optimisation (SOO) and buy-and-hold strategies. Our results show that the proposed approach significantly improves the cumulative return compared to SOO, from 14.29% to 62.04%, while also outperforming the buy-and-hold strategy.**

***Index Terms*—Directional changes, Genetic programming, Algorithmic trading, Multi-objective optimisation, technical analysis.**

I. Introduction

Algorithmic trading aims to exploit computer programs and algorithms to execute trades on financial markets. Such algorithms are typically designed to analyse market data and trade based on a set of predefined rules and parameters. The efficient market hypothesis (EMH) [9] argues that financial markets are "informationally efficient", in the sense that, at any given time, prices reflect all available information on an asset. According to the EMH, it is impossible to beat the market by using publicly available information and consistently achieving higher returns. Algorithmic trading can be seen as an attempt to gain an edge in the market by using advanced technology and large amounts of data to make trades, despite EMH implying that this should not be possible. To this aim, researchers in financial forecasting have explored a large arsenal of computational methods and techniques.

Most financial forecasting relies on the physical time scale, that is, by focusing on hourly, daily, or weekly data. Such an approach has been followed in numerous studies (e.g., see [6], [10]), but it is not always without shortcomings. In particular, it leads to non-continuous time series and potentially omits important information that is not reflected in the regular physical time intervals; consider, for example, a rapid and temporary price decline that will not be captured on daily data.

As a means to ameliorate such drawbacks of fixed time intervals, one can follow an alternative approach by focusing instead on key events. A relatively recent such technique is Directional Changes (DC) which identifies a series of crucial events defined by price movement over a user-defined threshold [11]. By setting a higher threshold we end up with fewer, but more significant, events, while selecting a lower threshold generates more, but perhaps less significant, ones. DC provides a complementary approach to fixed physical time intervals and allows the trader to obtain different sets of key events based on the same dataset. A more detailed discussion of the DC framework can be found in Section II-A.

Another challenge faced by traders is how to find the sweet spot in the trade-off between risk and profit in financial forecasting. As the perfect model does not exist, traders frequently have to make decisions based on a model that could discover more trading opportunities, or, reversely, that could reduce trading risk. Aggressive traders would opt for the former approach, while risk-averse traders for the latter one. A suitable approach to address this challenge is the use of multi-objective optimisation (MOO). MOO operates by finding a series of non-dominated sets of candidate solutions (also called Pareto fronts) that provide meaningful information on which candidate is better.

Previous work has provided evidence that DC is competitive compared to the physical time paradigm and has the potential to outperform it using GP-based algorithms [15]. This paper proposes a multi-objective genetic programming approach combining DC-based indicators and technical indicators based on physical time. The MOO approach we used relies on the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [8] and simultaneously optimizes each objective without being dominated by any other solution. In particular, we calculate 28 different DC indicators proposed by [3] and 28 common technical indicators presented in [13]. Together with these DC and physical-time indicators, the proposed trading strategy combines GP and MOO with DC. We run the trading strategies over 110 stocks from ten different international markets in six countries; these are Dow Jones Industrial Average (DJIA), Nasdaq Stock Market (NASDAQ), New York Stock Exchange (NYSE), Russell 2000 Index, and Standard and Poor's 500 (S&P500) in the United States, Nifty Fifty (NIFTY 50) in India, Taiwan Stock Exchange Corporation (TSEC) in China (Taiwan), DAX performance index in Germany, Nikkei 225 in Japan, and the Financial Times Stock Exchange 100 Index in the United Kingdom.

Building on the work of [15], we investigate whether MOO can further improve the performance of the GP-based strategy on financial forecasting. To assess whether our goal was achieved, we benchmarked MOO with a trading strategy combining single-objective optimisation and GP, namely SOO. The fitness functions used in MOO are cumulative return, risk, and risk-adjusted return, while the fitness function of SOO is a risk-adjusted return. As evaluation metrics, we use MOO's fitness functions, i.e., cumulative return, rate of return (ROR), and risk.

This paper is structured as follows. Section II presents a brief introduction to DC and multi-objective GP, while also discussing related work. In Section III, we discuss in detail the framework of the multi-objective GP and present the corresponding trading strategy. Section IV focuses on the experimental setup; we review the data collection and discuss the chosen benchmarks as well as the parameter tuning process. In Section V, we present and discuss our experimental results, while, finally, Section VI concludes the findings of this paper while also identifying future research directions.

II. Literature review and background

In this section, we introduce the background of the DC and multi-objective GP. Then, some relevant previous work will be presented.

A. Directional change

First proposed by [11] and formally defined by [21], DC is a novel way of summarising market price movement as a series of events. Therefore, the unit of reference for DC is an event, while the corresponding unit of the fixed-interval-based approach is a time period, e.g. minute, day, or week. The rationale behind the introduction of the DC framework is to discard events that do not substantially alter the landscape and focus only on key events. In particular, under the DC framework, the market is divided into upward and downward events. The start of a DC event is detected only when the price movement exceeds a given threshold, while the endpoint of the DC event is considered a confirmation point. The choice of the threshold is important, as the same dataset will lead to identifying different key events for different threshold values; higher thresholds lead to fewer events. Once confirmed, a DC event is frequently followed by an overshoot (OS) event; the OS event lasts until the price starts moving in the opposite direction, where eventually we obtain a new DC event. Therefore, any event belongs to one of the following types: upward DC event, upward OS event, downward DC event, and downward OS event. However, recent research shows that an OS event might be unnecessary, that is, a DC event could lead directly to another DC event [1].

B. Multi-objective GP

Genetic programming, proposed in [14], is an evolutionary computation algorithm that simulates Darwin's principle of natural selection. Similar to a genetic algorithm (GA), GP starts with initialising a population of individuals, where each individual represents a candidate solution for the problem we are facing and is evaluated through a fitness function. The process runs along several rounds (or, generations), and, from one generation to the next one, pairs of individuals will exchange part of their structure with each other, through the crossover operation. In this process, to achieve evolution and mimic natural selection, the better candidate, that is an individual with a better fitness function, has a higher probability to be involved in crossover. Meanwhile, the mutation operation randomly replaces part of an individual, mimicking the same process in nature. Using crossover and mutation, the population of individuals could ensure enough flexibility and lead to improvement over generations. Eventually, GP has the potential to find the global optimisation point.

GP and GA rely on the same fundamental processes and their difference is that the individuals of GP consist of functions such as a tree structure of actions and values, while the individuals of GA consist of raw data such as numbers. This leads to GP providing greater flexibility, at the expense of increased complexity, and potential to the population individuals to solve the target problem (e.g. see [2], [20]). Based on the specific research question at hand, we could define different functions. As a drawback, however, GP requires a better understanding of crossover and mutation operators, as these now apply to more complex individuals.

The fitness function, which evaluates each candidate solution, is the most critical factor for GP-based algorithms. In the domain of financial forecasting, examples of potential fitness functions include profit, rate of return, risk, or other more complex metrics such as the Sharpe ratio. However, in most cases, a single fitness function does not suffice to capture the full extent of the target problem. Consider, for instance, a setting where an increase in profit comes at a similar increase in risk. While previous work has considered the Sharpe ratio, a metric calculated by both risk and return, as a means to tackle this challenge under a GP-based trading strategy (see, e.g. [15], [7]), the results suggest a mediocre, arguably even bad, performance of the corresponding GP-based model leading to an average Sharpe ratio of 0.34.

In our work, we focus on multi-objective GP that uses at least two fitness functions for evaluation through the generations. In such a setting, multiple conflicting objectives need to be maximised or minimised [23]. A useful concept in this multi-objective setting is that of domination. We say that one solution dominates another when each fitness function of the first solution is at least as good as the second one, and at least one fitness function of the first solution is strictly better than the second one; otherwise, we say that these solutions are non-dominated. In other words, no solution could improve its performance against one objective function without reducing its efficiency with respect to another objective function. A series of non-dominated sets form the Pareto front and the ultimate goal of a multi-objective optimisation algorithm is to identify this front. The existence of many interrelated factors in the field of financial forecasting and the multiple conflicting objectives lead to a vast and complex search space, making the search for the Pareto front harder. Due to its inherent evolutionary method, GP-based approaches that simultaneously evolve a population of solutions make it easier to identify the set of non-dominated solutions among all complex potential solutions.

C. Related work

As discussed in the previous subsection, the concept of GP was first introduced by Koza [14] and has been frequently exploited in the domain of financial forecasting for several years [5]. To name a concrete example, Potvin et al. [19] applied GP for automatically generating short-term trading rules suitably adjusted for 14 Canadian companies listed on the Toronto stock exchange market. Their work demonstrated that these short-term trading rules were profitable when the market was stable or falling, otherwise, the buy-and-hold strategy worked better. Further examples include [4] that presented technical trading rules, based on GP, that performed better than the buy-and-hold trading strategy on S&P500 when considering also the transaction costs. Moreover, Myszkowski and Rachwalski [16] presented two decision trees, one for decisions to buy and one for decisions to sell. The experiments were tested on the Foreign Exchange market based on the technical and fundamental indicators. Although the risk was high, their approach was able to generate a trading strategy better than the buy-and-hold strategy. Very recently, [15] proposed a GP-based trading strategy, using the Sharpe ratio as a fitness function and financial data from 2015 to 2022, under the DC framework. Their results suggest that their approach could lead to high profit (13%) in a bull market and outperformed the buy-and-hold trading strategy (-4%).

The papers above used a single fitness function in the GP framework. As previously argued, such an approach is not always suitable and, hence, the performance of multi-objective GP has also been investigated. In particular, Hassan [12] studied multi-objective GP in a real dynamic world. They introduced the robustness metric to quantify the degree of robustness when multi-objective GP is applied in out-of-sample datasets; the results showed a significant improvement in the solution's robustness. More recently, Pimenta et al. [17] created an automated system including multi-objective optimisation, GP, technical analysis, and feature selection. They evaluated the performance of the system in six BOVESPA shares for two periods, from 2013 to 2015 and from 2015 to 2016 and they observed that their system returned profit even when the asset was devalued.

III. Methodology

This section presents the proposed algorithm and the trading strategy we used. For the proposed algorithm, we describe the terminal set and the function set we use, we discuss how the model is represented, we define the fitness functions we use and also discuss details of the GP. As discussed in the Introduction, we follow the framework of [15], so that we can compare the effectiveness of our multi-objective optimisation approach against the single-objective approach used therein.

A. Multi-objective Genetic programming model

1) Terminal set: The terminal set comprises the terminals of the GP tree and relies on the input data that will be used during GP training. We use 28 technical indicators under physical time series and 28 indicators under the DC framework as the terminal nodes of the proposed multi-objective genetic programming algorithm; these indicators can be found in [3] and [13]. In addition, we also use an Ephemeral Random Constant (ERC) which, when called, returns a random number from $\mathscr{U}_{[-1,1]}$, that is, the uniform distribution from -1 to 1. All 56 indicators (28 technical and 28 DC ones) have been normalised in the range from -1 to 1 to fit the ERC range.

The 56 indicators are outlined below; note that some of them are parameterised based on a time window, with each such time window leading to a new indicator. For DC-based indicators, we use the total price movement value (TMV), the overshoot value (OSV), the time-adjusted return of DC (R_{DC}), and the time spent on a trend (T_{DC}), for which there is no time parameter. We furthermore, use the total number of DC events (N_{DC}) over a selected time period, the time-independent coastline, that is the sum of the absolute values of TMV over a selected period (C_{DC}), and the difference between time spent on up trends and down trends in a time interval (A_T); each of these is computed over time windows of 10, 20, 30, 40, and 50 days. Finally, we use the averages of overshoot value (Average OSV), time-adjusted return (Average R_{DC}), and time spent on a trend (Average T_{DC}); each of these for time windows of 3, 5, and 10 days.

For the 28 technical indicators, we use Moving Average, Commodity Channel Index, Relative Strength Index, and William's %R; each of these is calculated over time windows of 10, 20, 30, 40, and 50 days. Also, we use Average True Range, and Exponential Moving Average with time windows of 3, 5, and 10 days, while we additionally use, without time windows, On Balance Volume and parabolic SAR. After including the ERC as well, we end up with 57 different terminals.

2) Function set: The function set is a rather simple, but expressive, one and includes two logical operators, namely AND and OR, and two relational operators, namely less than (<) and greater than (>). This choice is guided by the selection of indicators forming the terminal set.

3) Model representation: The multi-objective GP evolves a tree-based strategy to decide on the buy action in the proposed trading strategy. When each condition of an If-Then-Else (ITE) statement is satisfied, the tree will return a buy signal; see Part 1 of Figure 1 for an example. Therefore, in this ITE statement, an output of '1' corresponds to a buy action, while an output of '0' to a hold action; we remark that sell actions will be discussed in Section III-B.

To ensure the tree is valid, we require the root of part 1 to be an AND logical operator, while part 2 of the ITE statement is not included in the multi-objective GP algorithm because its values are constants, either 0 or 1, and there is no need to evolve them. Furthermore, to guarantee that both DC-based and physical-time indicators exist in the tree, we require that one branch connected to the root contains only DC-based indicators, while the other branch only physical-time based ones.

4) Fitness function: In this multi-objective GP trading strategy, we use three fitness functions: cumulative return, risk, and risk-adjusted return. The cumulative return and the risk are calculated as the sum and the standard deviation of the rate of return (ROR). The risk-adjusted return is defined as:

$$\mathsf{Adjusted\,return} = \frac{\mathsf{E}(R) - R_f}{\sqrt{\mathsf{Var}(R)}}, \tag{1}$$

where E and Var denote the sample mean and variance, respectively, R denotes the rate of returns, while R_f is the risk-free rate. To compute the fitness functions, we use the trading algorithm outlined in Section III-B, which indicates when the selling of the stocks will take place.

5) Selection method and operators: We use elitism, subtree crossover, and point mutation. Regarding mutation, functions can only mutate to their paired function, that is AND to OR and vice versa and, similarly, > to < and vice versa, DC-based indicators can only mutate to another DC-based indicator, and similarly for a physical-time indicator. Furthermore, cross-over is carefully designed to guarantee the validity of the resulting program, e.g., by enforcing that one branch has only DC-based indicators and the second branch only physical time-based ones. We also use tournament selection to choose individuals as parents for the above operators. A summary of the GP configuration is presented in Table I.

TABLE I
CONFIGURATION OF THE GP ALGORITHM

Configuration	Value
Function set	AND, OR, >, <
Terminal set	28 DC indicators, 28 physical time indicators, and ERC
Genetic operators	Elitism, subtree crossover, and point mutation
Selection	Tournament

6) Multi-objective optimisation: In our work, we focus on evaluating more than one fitness function and we rely on NSGA-II (Non-dominated Sorting Genetic Algorithm) for this. As discussed in Section II-B, a candidate solution dominates another candidate solution when the fitness function of the former solution is as good as that of the latter solution and at least one fitness function of the former solution is strictly better. With this in mind, NSGA-II works as follows: First, it divides the population into a series of different fronts, so that solutions within the same front are not dominated by each other, while any solution in a front earlier in the series dominates any solution in a later front. Then, NSGA-II sorts candidate solutions within each front, using a single fitness function for the same front. As a next step, the crowding distance is calculated as the absolute normalised difference between two neighbouring solutions. It is important to note that the values for the first and last solutions are set to infinity. In this way, NSGA-II can sort candidate solutions based on the front's order and the crowding distance's value. The final result arising from this process is the one with the maximum crowding distance, i.e. the first solution in the first front.

B. Trading strategy

The trading strategy we use boils down to answering the question "Is the stock price going to increase by r% within the next n days?". The goal of the multi-objective GP tree is to provide an answer to this question.

When the GP tree returns '1' as output, we buy one amount of stock unless we already own the stock. Otherwise, when the GP tree returns '0', we take no action (i.e., we hold). For the case of stocks that we own, if their price increases by at least r% in the next n days, then we sell the stock on the day this threshold is exceeded, otherwise, we sell the stock on the final day, that is, day n. We remark that, by design, it is not possible to sell a stock that we do not already own (that is, short-selling is not allowed). When a sell action takes place, we calculate and record the profit obtained; we assume a transaction cost of 0.025% per trade. The above trading strategy is summarised in Algorithm 1.

The rate of return from each trade is computed based on the price P_b we bought and the price P we sold the stock; see Equation (2). After creating a list with all returns from all trades executed, we compute the sample mean, which gives the overall rate of return; this, in turn, is provided as input to Equation (1) to determine the adjusted

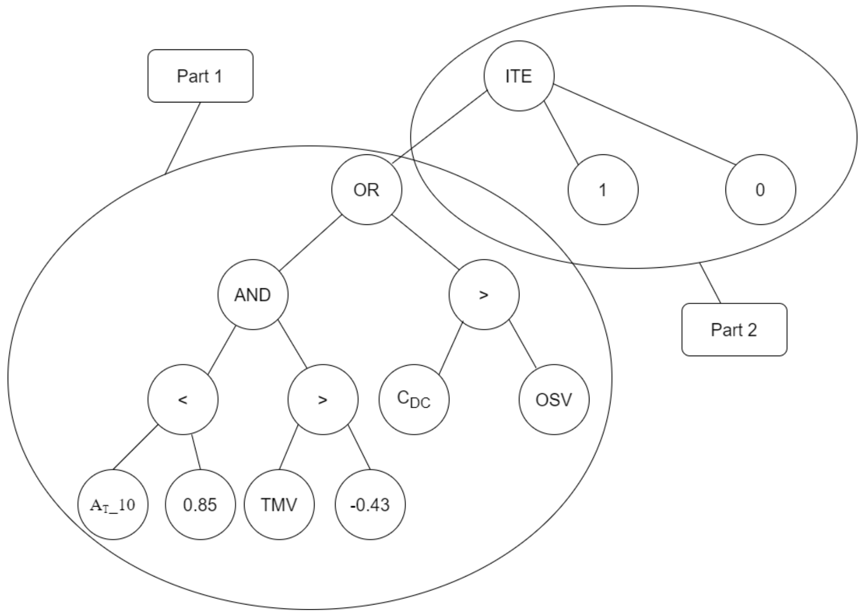

Fig. 1. An example of the GP tree and the If-Then-Else structures. If the AT for 10 days is less than 0.85 and TMV is greater than −0.43 or C_{DC} is greater than OSV, then we get a signal for a buy action; otherwise, we hold.

Algorithm 1 The trading strategy given threshold r% and duration n days

---

Require: Initialise variables (O represents the prediction of the GP tree, while $index$ indicates whether the stock is held)

if $O = 1$ and $index = 0$ **then**
  Buy one amount of stock
  $index \leftarrow 1$
  $N \leftarrow i$ //Starting time for trade: i is always the current time
  $K \leftarrow p$ //Stock price when buying: p is always the current price
else
  if ($index = 1$ and $p > (1 + r/100) \times K$) OR $(i - K) > n$ **then**
    Sell the stock
    $index \leftarrow 0$
    Calculate and record profit
  end if
end if

---

return fitness function. The *risk* is defined as the standard deviation of that list of returns, see also Equation (3), while the cumulative return is the sum of the rate of return for each trade.

$$R = \left\{ \frac{0.99975 \cdot P - 1.00025 \cdot P_b}{1.00025 \cdot P_b} \right\} \cdot 100\% \tag{2}$$

$$\text{Risk} = \sqrt{\text{Var}(R)} \tag{3}$$

IV. Experimental setup

In this section, we will present the experimental setup including the data, benchmarks, and parameter tuning process.

A. Data

The proposed multi-objective GP trading strategy is applied to the daily historical data from 10 international markets in 6 countries, namely the Dow Jones Industrial Average (DJIA), the Nasdaq Stock Market (NASDAQ), the New York Stock Exchange (NYSE), the Russell 2000 Index, and the Standard and Poor's 500 (S&P500) in the United States, the Nifty Fifty (NIFTY 50) in India, the Taiwan Stock Exchange Corporation (TSEC) in China (Taiwan), the DAX performance index in German, Nikkei 225 in Japan, and the Financial Times Stock Exchange 100 Index in the United Kingdom. From each market, we select at random ten stocks as well as the market index, and we consider the time window between 24.11.2010 and 23.11.2020; this gives 2517 days. Overall, we obtained 110 datasets; 10 stocks and the index for each of the 10 markets. We split each dataset into three parts, that is, training (60%), validation (20%), and testing (20%), and we use the training and validation sets for parameter tuning. Our final experiment is trained on a combined dataset (training and validation) and tested against the test set. We used the available data to compute the 56 indicators as explained in Section III-A1.

B. Benchmarks

In order to investigate whether multi-objective optimisation could improve the GP-based strategy combining technical and event-based indicators, we compare the proposed multi-objective GP trading strategy with the other two approaches: the single objective GP trading strategy (SOO) and the buy and hold strategy (B&H). B&H is a passive trading strategy that allows traders to buy a stock and hold it regardless of the fluctuation of the stock. Once selecting a stock, the trader who uses the B&H strategy is always concerned with the long-term profit of the stock and ignores the price movement in the short term. SOO is considered by [15] and has the same configuration as our proposed multi-objective GP trading strategy, except it only applies a single (risk-adjusted) fitness function.

C. Parameter tuning for GP

We performed a grid search to determine the optimal GP parameters for the two GP-based algorithms, and tuning took place based on 50 runs of ten randomly selected datasets and using the validation set. Eventually, we selected the parameter pair with the best average performance over these 50 runs. Based on [18], we adopted the most common values for each parameter, namely 4, 6, and 8 for the max depth; 100, 300, and 500 for the population size; 0.75, 0.85, and 0.95 for the crossover probability; 2, 4, and 6 for the tournament size; and, finally, 25, 35, and 50 for the number of generations. As we set the mutation probability to be equal to (1-crossover probability), we did not need to consider this parameter. Table II shows the selected parameters and their chosen values after tuning.

TABLE II
PARAMETERS OF THE GP ALGORITHM

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

D. Parameter tuning for the trading strategy

We remark that there are three parameters involved in our trading strategy. Two of them (r and n) are related to the question of whether the stock price will increase by r% during the next n days, while the third parameter is the threshold used to detect events in the DC framework. Rather than tuning the above parameters and selecting the best set across all datasets (which is what we did for the parameters of the GP-based algorithms), we decided to allow for tailored values for each dataset. To obtain these values, we conducted experiments using various values for these parameters and kept the best-performing values for each dataset. Table III presents the configuration space for these three parameters.

TABLE III
CONFIGURATION SPACE FOR THE TRADING STRATEGY

Parameters	Configuration space
n (length of the trading window)	1, 5, 15
r (percentage of price movement)	1%, 5%, 10%, 20%
Threshold of DC	0.001, 0.002, 0.005, 0.01, 0.02

V. RESULT AND ANALYSIS

We now present and discuss the results of our experimental study. We begin by comparing the multi-objective optimisation GP-based approach to the single-objective one (see Section V-A) and continue by contrasting the two GP-based approaches to the buy and hold strategy (see Section V-B). We remark that, as discussed in Section III-A, for MOO we report the result indicated by NSGA-II using the maximum crowding distance.

A. Overall

Table IV presents summary statistics across 110 datasets for the performance of multi-objective (MOO) and single-objective optimisation (SOO) GP-based algorithms and the metrics of cumulative return, rate of return, and risk. The values reported for the two GP-based algorithms are the average of 50 independent runs.

From Table IV, it is clear that the proposed MOO-GP strategy outperforms SOO-GP in terms of all evaluations of cumulative return and rate of return except for standard deviation. In contrast, SOO-GP performs better in the rest of the assessments. A direct comparison shows that using multi-objective optimisation significantly increases the average cumulative return; an almost 4-fold increase from 14.30% to 62.05% cumulative return. A plausible explanation for this behaviour is that MOO significantly improved profit by sacrificing the risk, from 0.0590 to 0.0778. Considering the massive improvement in the profit, it can be argued that the slight increase in risk is worth it.

TABLE IV
SUMMARY STATISTICS OF THE GP-BASED ALGORITHMS. THE BEST VALUES BETWEEN MOO AND SOO PER METRIC APPEAR IN BOLDFACE.

Measurement	Cumulative return		Rate of return		Risk	
Algorithms	MOO	SOO	MOO	SOO	MOO	SOO
Average	**62.05**%	14.30%	**2.75**%	0.91%	0.0778	**0.0590**
Median	**44.62**%	10.85%	**2.70**%	0.77%	0.0651	**0.0480**
Standard deviation	0.6838	**0.3778**	0.0208	**0.0125**	0.0513	**0.0357**
Maximum	**374.22**%	186.62%	**9.90**%	6.00%	0.3226	**0.2413**
Minimum	**-53.17**%	-229.56%	**-1.92**%	-3.34%	0.0116	**0.0097**

Figure 2 presents the box plot of the above result. To have a clearer view, we zoomed in on the 'box' in the plot. From Figure 2, we can reach a similar conclusion to Table IV, that is, the MOO strategy has better cumulative return and ROR than SOO, while the risk of MOO is slightly worse. Besides, the bottom line of both box plots is higher than zero in terms of cumulative return. In other words, both MOO and SOO strategies lead to positive profit in more than 75% of the datasets. Furthermore, the values and overall box plot of the MOO strategy are higher in terms of ROR and cumulative return when compared to the SOO strategy. When arguing about risk, the MOO's plot is still higher than the SOO's one, indicating again a more risky behaviour by MOO.

To confirm the above results, we apply the non-parametric Kolmogorov-Smirnov (KS) test to support the

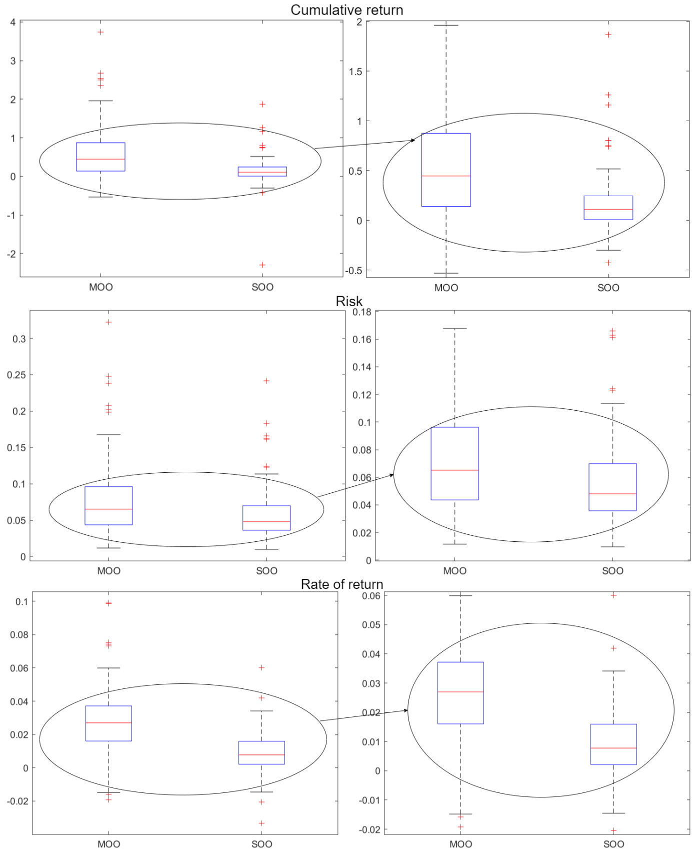

Fig. 2. Box plot of MOO and SOO

analysis for the two algorithms. We performed the test for each metric between the MOO and SOO strategies. When the p-value for a metric is below 0.05, the null hypothesis is rejected at the 5% significance level. The p-values for cumulative return, ROR, and risk are 1.1709e-10, 1.0067e-13, and 0.0055, respectively. Therefore, the difference between the distributions of the MOO and SOO strategies in terms of all three metrics is statistically significant. On the positive side, this indicates that the MOO strategy significantly outperformed the SOO strategy in terms of cumulative return and ROR. On the other hand, however, the risk of the SOO strategy outperformed that of the MOO strategy.

B. Buy and hold

In the previous section, we compared the proposed MOO strategy with the SOO one and concluded that MOO has a significantly higher cumulative return and ROR with worse risk than SOO. In this section, we will compare both of them to the B&H strategy in cumulative return. The reason we are not doing the comparison on the ROR and risk is that these are undefined for B&H. Indeed, the B&H approach involves just one trade, i.e., buying a single unit of stock on the first day and selling it on the last day. Thus, the comparison is focused on the cumulative return.

From Table V, we can observe that B&H leads to a highly volatile result as the standard deviation is 1.8051; this happens because the B&H strategy relies heavily on the historical price movements of the stock itself. When the stock is in an uptrend, the B&H approach is able to produce extremely high returns, such as the 1753.05% in Table V. Although the proposed strategy could not beat the maximum cumulative return of the B&H method, it has the highest average, median, and minimum cumulative return among the three approaches.

TABLE V
CUMULATIVE RETURN OF THE GP-BASED AND B&H STRATEGY. THE BEST VALUES AMONG THE THREE ALGORITHMS PER METRIC APPEAR IN BOLDFACE.

Measurement	Cumulative return		
Algorithms	MOO	SOO	B&H
Average	**62.05%**	14.30%	41.11%
Median	**44.62%**	10.85%	11.44%
Standard deviation	0.6838	**0.3778**	1.8051
Maximum	374.23%	186.62%	**1753.05%**
Minimum	**-53.17%**	-229.56%	-89.62%

Figure 3 presents the distribution of the MOO, SOO, and B&H strategies. Compared to the B&H strategy, MOO has a higher box plot, and the SOO strategy has a thinner box plot, indicating a higher cumulative return and better risk. In other words, the MOO strategy can find the model that leads to a higher return than the B&H strategy, while the SOO tends to find the model with less risk.

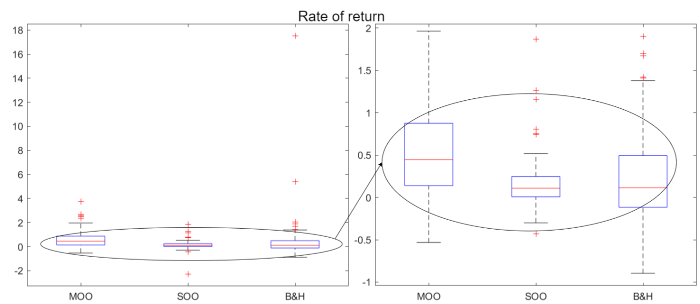

Fig. 3. Box plot for MOO, SOO, and B&H strategy

Similarly, the non-parametric KS test is run between MOO with SOO and B&H strategy to prove the above result. Table VI presents the p-value of the statistical test. Considering the Bonferroni correction, the null hypothesis is rejected at the 5% significance level when the p-value is below 0.025, as the B&H strategy is chosen to be the target algorithm that is compared against the other two strategies [22]. We can observe that the p-value of the MOO strategy is below 0.025, indicating that the difference between the MOO and B&H strategy is statistically significant. Combined with the above observation, we can claim that the proposed MOO strategy outperforms the B&H strategy. Besides, although the SOO has a stable cumulative return, the difference between SOO and B&H strategy is statistically insignificant.

TABLE VI
THE p-VALUES BETWEEN B&H STRATEGY AND EACH OF THE MOO AND SOO STRATEGIES; p-VALUES BELOW 0.025 APPEAR IN BOLDFACE.

P-value	B&H
MOO	1.1672e-08
SOO	1.1542e-01

VI. CONCLUSION

This paper presents and evaluates a multi-objective optimisation GP strategy combining physical and event-based indicators. We have investigated the performance of the proposed strategy under 110 stocks from 10 different markets. To evaluate the efficiency of our approach, we have used as benchmarks a single objective optimisation GP strategy and the well-known buy-and-hold strategy. The experiments have shown that the proposed strategy significantly increased cumulative return from 14.30% to 62.05% when compared to the single objective strategy. Furthermore, the statistical tests provided evidence that the proposed strategy outperformed the buy-and-hold strategy. Although the risk of the proposed multi-objective strategy was also greater, we argue that the gain in the cumulative return is sufficient to tolerate this risk increase.

Since our approach provides a high return, future work will focus on providing similar return guarantees while simultaneously reducing the risk by adjusting the fitness function. In addition, there still exist interesting future research questions on combining physical and DC methods. While in this paper we only used technical and DC indicators to train the GP directly, in future work we could have different ways to combine the physical and event-based methods. One potential approach would be to incorporate both technical and DC indicators on an equal basis in our model; note that our GP-based model was trained by the total return and risk, which could be contributed mainly by technical or DC indicators.

REFERENCES

[1] Adegboye, A., Kampouridis, M.: Machine learning classification and regression models for predicting directional changes trend reversal in fx markets. Expert Systems with Applications **173**, 114645 (2021)

[2] Al-Sahaf, H., Bi, Y., Chen, Q., Lensen, A., Mei, Y., Sun, Y., Tran, B., Xue, B., Zhang, M.: A survey on evolutionary machine learning. Journal of the Royal Society of New Zealand **49**(2), 205–228 (2019)

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

[4] Becker, L.A., Seshadri, M.: Gp-evolved technical trading rules can outperform buy and hold. In: 3rd international workshop on computational intelligence in economics and finance. Citeseer (2003)

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

[6] Cao, L., Tay, F.E.: Financial forecasting using support vector machines. Neural Computing & Applications **10**(2), 184–192 (2001)

[7] Christodoulaki, E., Kampouridis, M.: Using strongly typed genetic programming to combine technical and sentiment analysis for algorithmic trading. In: 2022 IEEE Congress on Evolutionary Computation (CEC). pp. 1–8 (2022)

[8] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation **6**(2), 182–197 (2002)

[9] Fama, E.F.: Efficient capital markets: A review of theory and empirical work. Journal of Finance **25**(2), 383–417 (1970)

[10] Garcia-Almanza, A.L., Tsang, E.P., et al.: Forecasting stock prices using genetic programming and chance discovery. In: 12th International Conference On Computing In Economics And Finance. No. 489 (2006)

[11] Guillaume, D.M., Dacorogna, M.M., Davé, R.R., Müller, U.A., Olsen, R.B., Pictet, O.V.: From the bird's eye to the microscope: A survey of new stylized facts of the intra-daily foreign exchange markets. Finance and stochastics **1**(2), 95–129 (1997)

[12] Hassan, G.N.A.: Multiobjective genetic programming for financial portfolio management in dynamic environments. Ph.D. thesis, UCL (University College London) (2010)

[13] Kelotra, A., Pandey, P.: Stock market prediction using optimized deep-convlstm model. Big Data **8**(1), 5–24 (2020)

[14] Koza, J.: On the programming of computers by means of natural selection. Genetic programming (1992)

[15] Long, X., Kampouridis, M., Kanellopoulos, P.: Genetic programming for combining directional changes indicators in international stock markets. In: International Conference on Parallel Problem Solving from Nature. pp. 33–47. Springer (2022)

[16] Myszkowski, P.B., Rachwalski, Ł.: Trading rule discovery on warsaw stock exchange using revolutionary algorithms. In: 2009 International Multiconference on Computer Science and Information Technology. pp. 81–88. IEEE (2009)

[17] Pimenta, A., Nametala, C.A., Guimarães, F.G., Carrano, E.G.: An automated investing method for stock market based on multiobjective genetic programming. Computational Economics **52**(1), 125–144 (2018)

[18] Poli, R., Langdon, W.B., McPhee, N.F.: A field guide to genetic programming. Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk` (2008), (With contributions by J. R. Koza)

[19] Potvin, J.Y., Soriano, P., Vallée, M.: Generating trading rules on the stock markets with genetic programming. Computers & Operations Research **31**(7), 1033–1047 (2004)

[20] Raghavjee, R., Pillay, N.: A comparison of genetic algorithms and genetic programming in solving the school timetabling problem. In: 2012 Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC). pp. 98–103 (2012)

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

[22] Weisstein, E.W.: Bonferroni correction. https://mathworld. wolfram. com/ (2004)

[23] Zhao, H.: A multi-objective genetic programming approach to developing pareto optimal decision trees. Decision Support Systems **43**(3), 809–826 (2007)