Multi-Objective Genetic Programming-based Algorithmic Trading, using Directional Changes and a Modified Sharpe Ratio Score for Identifying Optimal Trading Strategies

Xinpeng Long[1*], Michael Kampouridis[1] and Tasos Papastylianou[1]

[1*]School of Computer Science and Electronic Engineering, University of Essex, Street, Colchester, CO43SQ, Essex, UK.

*Corresponding author(s). E-mail(s): xl19586@essex.ac.uk;
Contributing authors: mkampo@essex.ac.uk;
tasos.papastylianou@essex.ac.uk;

Abstract

This study explores the integration of directional changes (DC), genetic programming (GP), and multi-objective optimisation (MOO) to develop advanced algorithmic trading strategies. Directional changes offer a dynamic, event-based approach to market analysis, identifying significant price movements and trends. Genetic programming evolves trading rules to discover effective and profitable strategies. However, financial trading presents a multi-objective challenge, balancing conflicting objectives such as returns and risk. We propose a novel algorithmic trading framework, termed MOO3, which integrates genetic programming with the NSGA-II multi-objective optimisation algorithm to optimise three fitness functions: total return, expected rate of return, and risk. While the use of NSGA-II itself is well-established, our contribution lies in how we apply it within a trading context that combines (i) directional changes, (ii) genetic programming with both DC-based and physical-time indicators, and (iii) a modified Sharpe Ratio for post-optimisation strategy selection based on trader preferences. Utilising indicators from both paradigms allows the GP algorithm to create profitable trading strategies, while the multi-objective fitness function allows it to simultaneously optimise for risk. A definitive strategy is chosen from Pareto-optimal solutions using the modified Sharpe Ratio, allowing traders to prioritise multiple objectives. Our methodology is tested on 110 stock datasets from 10 international markets, aiming to demonstrate that the multi-objective framework can yield superior trading strategies with lower risk. Results indicate that the MOO3 algorithm consistently and significantly outperforms single-objective optimisation (SOO) methods, even when the same SOO criterion is employed for choosing a single, definitive investment strategy from the Pareto front.

Keywords: Directional changes, Genetic programming, Algorithmic trading, Multi-objective optimisation

1 Introduction

Algorithmic trading has attracted increased attention from investors over the last few years. The adaptability of the market to new algorithms has led to the constant pursuit of novel and efficient trading strategies. One such algorithm is directional changes (DC), which is an event-based technique that summarises physical time series into a series of significant events. These significant events are defined as the price movement over a user-defined threshold θ, such as a price change of 0.5%. Under the DC framework, the market is divided into uptrend and downtrend segments, each characterised by ‘directional change’ (DC) and ‘overshoot’ (OS) events. The DC event is defined as the span from the beginning of a trend segment up until the price moves by at least θ in the trend’s direction. Following this, the OS event is then defined as the remainder of that segment; the start of the next observed DC event marks the end of the previous segment, and the start of the new one where the trend direction has reversed.

In addition, Genetic programming (GP), which belongs to the family of evolutionary algorithms, has shown great promise in developing effective trading strategies (Brabazon et al, 2020). GP evolves populations of trading rules or models through processes analogous to natural selection. This evolutionary approach enables the discovery of potentially profitable trading strategies that might not be evident through traditional methods, such as strategies derived from comparing the performance of technical analysis indicators. GP's flexibility and adaptability make it particularly well-suited for the dynamic and complex nature of financial markets.

Combining directional changes with genetic programming can lead to the formulation of robust trading strategies (Long et al, 2022b; Long and Kampouridis, 2024). However, the financial markets present a multi-objective optimisation (MOO) challenge, where traders often need to balance conflicting objectives such as maximising returns and minimising risk. Multi-objective optimisation techniques provide a framework for simultaneously optimising such competing goals, yielding a set of Pareto-optimal solutions that offer various trade-offs. Ever since DC was proposed by Guillaume et al (1997), numerous studies have demonstrated its effectiveness in creating trading strategies that achieve high returns at relatively low risk (Bakhach et al, 2016b; Ao and Tsang, 2019; Salman et al, 2022).

Most existing DC-based trading strategies rely on single-objective optimisation methods or aggregate metrics such as the Sharpe ratio. These methods often oversimplify the trade-off between conflicting objectives like return and risk, potentially leading to sub-optimal strategies. Furthermore, while genetic programming has demonstrated effectiveness in evolving robust trading rules, previous studies typically focus on maximising returns or minimising risk independently, neglecting the simultaneous optimisation of both. Existing multi-objective optimisation (MOO) techniques in algorithmic trading tend to overlook the integration of event-based DC frameworks with evolutionary methods like GP, limiting their adaptability in volatile and dynamic market environments.

To address these limitations, this study proposes a novel framework for algorithmic trading, named MOO3 (Multi-Objective Optimisation with three objectives), which integrates directional changes, genetic programming, and multi-objective optimisation to develop and evaluate advanced trading strategies. The framework uses genetic programming (GP) in conjunction with the well-known NSGA-II algorithm (Deb et al, 2002) to simultaneously optimise three key fitness criteria: total return, expected rate of return, and risk. Although NSGA-II is a well-established method in the field of multi-objective optimisation, the novelty of this work lies in its tailored application within a trading context that brings together several distinct elements. Specifically, our approach uniquely integrates: (i) the directional changes (DC) paradigm, which offers a data-driven method for analysing price movements; (ii) genetic programming models that incorporate both DC-based features and conventional time-series indicators, thereby enriching the search space with diverse representations of market behaviour; and (iii) a modified version of the Sharpe Ratio that is used for selecting strategies after optimisation, enabling customisation based on trader-specific preferences for risk and return.

We run experiments on 110 datasets (stocks) from 10 different international markets. Our aim is to show that the utilisation of a multi-objective optimisation framework leads to more profitable trading strategies at low risk.

The rest of this study is organised as follows. Section 2 introduces the necessary background information on DC, Genetic Programming, and NSGA-II. Section 3 presents related work on directional changes and multi-objective optimisation in algorithmic trading, while Section 4 presents the methodology applied in this paper. Section 5 discusses the experimental setup, as well as the benchmarks and datasets used in this work. Section 6 is dedicated to presenting the findings from our experiments, and finally, Section 7 concludes this paper and discusses potential future work.

2 Background

This section provides background information on key concepts of our article, namely directional changes, NSGA-II and genetic programming.

2.1 Directional changes

The DC approach is an event-based approach for summarising market price movements. In contrast to the physical time approach, where time is divided into fixed intervals (e.g. daily closing price), DC summarises the price data into a series of upturn and downturn events, which are detected when the price moves over a user-defined *relative* threshold θ. That is, the time point at which the current price increases (decreases) by $\theta\%$ relative to the most recent minima (maxima), marks a new upturn (downturn) DC event, starting at the most recent minima (maxima) and ending at the current time point in question. Until the next downturn (upturn) DC event is observed, the rest of the uptrend (downtrend) is considered the overshoot (OS) event. In conclusion, the DC framework could be regarded as a series of downturn DC, downturn OS, upturn DC, and upturn OS events.

From the structure of DC, it becomes evident that the user-defined threshold holds a position of paramount importance within the DC algorithm. Different threshold values lead to different series of events for the same fixed-scale price series. Specifically, higher threshold values result in fewer but more significant DC events, whereas lower threshold values produce smaller and more frequent DC events. This is a very important property of the DC framework, since different traders may have different opinions as to what constitutes 'significant' as opposed to 'non-significant' events from their point of view. The presence of such a threshold, therefore, inherently provides the DC framework with the required flexibility, which allows traders to adapt the algorithm so as to best match their needs.

Figure 1 presents an example of how different thresholds could impact the same fixed-scale price series. The green line represents a series of DC and OS events when using a threshold of 1%, while the red line represents a series of DC and OS events using a threshold of 1.5%. The original price is denoted by the blue line. When looking at the red line, generated by the 1% threshold:

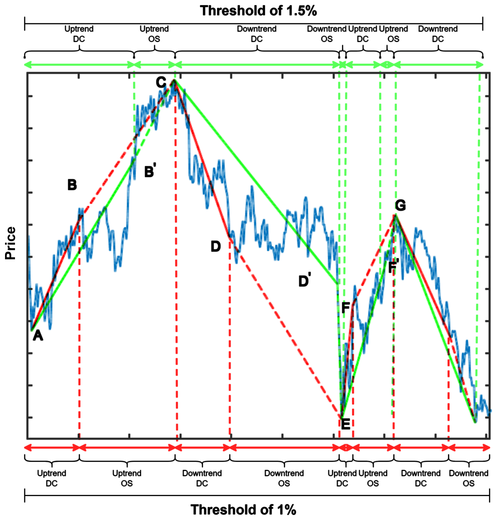


Fig 1 An example for DC. The blue line indicates the physical time series, the red line denotes a series of DC and OS events as defined by a threshold of 1%, while the green line denotes a series of DC and OS events as defined by a threshold of 1.5%. DC events are depicted with solid lines, while dotted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

- The interval between points A and C represents an *uptrend* segment, where AB represents the DC *upturn* event, and BC represents the *overshoot* (OS) event,
- The interval between C and E represents a *downtrend* segment, where CD represents the DC *downturn* event, and DE represents the OS event,
- The interval between E and G represents an *uptrend* segment, where EF represents the DC *upturn* event, and FG represents the OS event, and
- The interval beyond G represents a *downtrend* segment, with a DC and OS event.

Similarly, when looking at the green line, generated by the 1.5% threshold:

- The interval between points A and C represents an *uptrend* segment, where AB′ represents the DC *upturn* event, and B′C represents the *overshoot* (OS) event,
- The interval between C and E represents a *downtrend* segment, where CD′ represents the DC *downturn* event, and D′E represents the OS event,
- The interval between E and G represents an *uptrend* segment, where EF′ represents the DC *upturn* event, and F′G represents the OS event, and

- The interval beyond G represents a *downtrend* segment, with a DC event only.

Points B, B′, D, D′, F, and F′ are referred to as DC confirmation points, as these are the moments in time that one can confirm that a directional change has occurred. However, it is important to note that DC events can only be confirmed retrospectively, i.e. once the price has changed by at least θ. For example, point D is a downwards DC confirmation point, as it represents a change of θ relative to the previous tallest peak occurring at C. Before reaching point D, however, the trader could not have known they were in a DC downturn interval, but would instead have still been under the assumption that the market is in an upwards OS event, which started at point B.

A major advantage of the directional changes paradigm is that it allows traders to focus on meaningful market shifts (defined by the threshold θ), which reduces exposure to noise-driven trading risks. By focusing on looking at significant events rather than data under fixed-time methods, the algorithm adapts dynamically to different market conditions, hence mitigating risks associated with sudden price changes.

To conclude, directional changes is an alternative to physical time price summaries, which focuses on important events that occur in the market. Under DC frameworks, price movements can be viewed from a new (event-based) perspective that would not have been possible under physical time (Glattfelder et al, 2011).

2.2 Genetic programming

Genetic Programming (GP) is a bio-inspired technique that evolves computer programs to tackle problems or execute tasks. It incorporates key elements for an effective global search: (i) instead of focusing on a single solution, GP operates with a population of candidate solutions (individuals), (ii) the fitness function evaluates the quality of each individual in the population, favoring higher-quality solutions for progression to the next generation, and (iii) genetic operators explore the solution space by generating new offspring individuals through a stochastic selection process based on fitness.

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

```
GP($p$, $Fitness$, $pc$, $pm$)
    $p$: population size
    $Fitness$: determines the quality of solutions
    $pc$: crossover rate
Initialise population: $P \leftarrow$ Generate $p$ individuals (candidate solutions) at random
Evaluate: for each $i$ in $P$, calculate Fitness(i)
while termination condition not satisfied do
    $P_g \leftarrow$ Create new (empty) population for generation $g$
    Mating: probabilistically select $p$ individuals from $P$. For each of these:
      i. Toss a biased coin, with bias = $pc$
     ii. if Heads then perform crossover with another probabilistically selected
         individual
    iii. otherwise perform mutation on the selected individual
     iv. finally add the resulting offpsring to $P_g$
    Evaluate: for each $i$ in $P_g$, calculate Fitness(i)
    Survival: $P \leftarrow$ return the best $p$ individuals from the set $P \cup P_g$
end while
Return the individual with the highest fitness from $P$
```

Algorithm 1 outlines the high-level pseudocode of GP. It initiates a population of p candidate solutions, assessed by a fitness function. This fitness function is problem-specific. In each iteration (within the *while* loop), a new population emerges by probabilistically selecting fitter individuals from the current population. Some undergo crossover or mutation, introducing modifications for exploring the search space, while others are retained unchanged. This process continues until a maximum number of generations is reached or a (near-)optimal solution is found, serving as a termination condition. This evolutionary approach allows GP to conduct a robust global search in the candidate solution space, minimising the risk of being confined to local minima. Details about the specific GP we use in our experiments are given in Section 4.

2.3 NSGA-II

In the real world, there are plenty of problems that require a trade-off solution among multiple conflicting objectives. For instance, in the field of finance, traders seek a balance between profit and risk. They look for a trading strategy that combines high profit with low risk. In other words, they are solving a problem that maximises a profit objective and minimises a risk objective. It is vital to recognise that profit and risk represent two inherently conflicting objectives, where, traditionally, high profit is often associated with high risk, while low risk tends to correlate with reduced profit. Usually, traders consider profit and risk together, trying to achieve a good balance between the two. One way of tackling this problem is by using aggregate metrics; these involve combining all the different objectives into a single mathematical expression, to be optimised directly as a single objective. A common example of such an aggregate metric is the Sharpe ratio (Sharpe, 1994), defined as the ratio of the expected rate of return over the risk. Fusing multiple objectives in this manner can be appealing, as this can simplify the evaluation of the genetic algorithm considerably. It also easily allows one to specify 'weights' for each objective, denoting the extent to which an investor values each of the different components of the problem. However, using such a predefined 'recipe' for condensing multiple factors into a single number, risks oversimplifying the complex relationship which underlies the different objectives, and thus risks misrepresenting the performance evaluation of the investment portfolios in question during the algorithmic process. On the other hand, in a multi-objective optimisation (MOO) approach, the different objectives (e.g. both return and risk) are optimised independently.

NSGA-II (Non-dominated Sorting Genetic Algorithm II) (Deb et al, 2002) is a state-of-the-art genetic algorithm, effectively an extension of the traditional single-objective genetic algorithm approach, allowing one to efficiently tackle problems with multiple, potentially conflicting objectives. Its key innovation is the *non-dominated sorting* technique, which relies on the concept of Pareto dominance in order to enforce a ranking among the pool of potential candidate solutions (in the context of GP, these would reflect different investment strategies, for instance). A candidate solution is said to Pareto-dominate another solution if it is better in at least one objective, and not worse in any other objective. At any point during the course of a GP algorithm, the population of candidate solutions can therefore be divided into two subsets: a set of *dominated* 'individuals' (i.e. candidate solutions), for which there is at least one other solution which Pareto dominates the former; and a set of *non-dominated* individuals, where no such other dominating solution exists in the population. In this scenario, the subset of non-dominated individuals then naturally forms a kind of "outer boundary", whereby the 'inner' individuals that find themselves surrounded by this boundary are all effectively dominated by the points that make up the boundary. We call this boundary formed by non-dominated individuals the *Pareto front.* NSGA-II leverages this concept of a Pareto front, by partitioning the population into multiple fronts, each of which is ranked according to their order of non-dominance. In other words, if we define the primary front as having the topmost rank (i.e. rank 1), then we can obtain the next front (i.e. rank 2), by removing all the rank 1 individuals from the population, and recalculating the Pareto front for the remaining population. Repeating this process until all the individuals in the population have been allocated a rank, results in a hierarchical partitioning of the population into multiple fronts, whereby each front consists of individuals that Pareto dominate individuals in subsequent fronts. NSGA-II then simply uses this ranking as the main force driving the evolutionary process; this allows the genetic algorithm to favour non-dominated solutions over dominated ones during the *mating selection* process, while still maintaining diversity over generations by virtue of solutions being suitably distributed over the entire Pareto front. In order to ensure good coverage over the whole Pareto front in this manner, NSGA-II also introduces the concept of *crowding distance*, which encourages the spread of solutions by considering the density of Pareto optimal solutions around each candidate, and incorporating that information into the fitness function. The crowding distance is calculated as the normalised *Manhattan distance* between the two individuals closest to the solution of interest within the same front. This is effectively calculated as

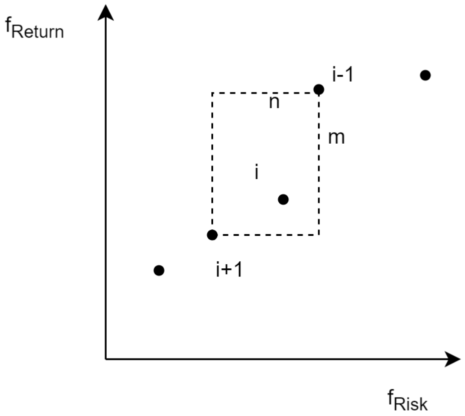


Fig 2 An example of crowding distance for a two-objective problem. The crowding distance of solution i here is defined as the Manhattan distance of its closest neighbours, denoted here as $i-1$ and $i+1$

the sum of absolute differences between the two individuals, across the various objectives. An example of the crowding distance calculation for a return-risk two-objective problem is shown in Figure 2, where points $i-1$ and $i+1$ are the two neighbouring points of point i within the same front. The generic formula for a crowding distance with k objectives is presented by Equation 1 below:

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

where $f_x(i)$ represents how well the i-th individual in the population performs with respect to a specific objective 'x'; $\max_j \left[f_x(j) \right]$ is the performance of the individual for which objective 'x' achieves its most favorable value (assuming a 'maximizing' fitness objective); and $\min_j \left[f_x(j) \right]$ corresponds to the performance of the individual with the least favourable value for 'x'. It is worth noting that the individuals with the best value on a particular objective are assigned the maximum crowding distance of 1 for that component, since they lack nearby individuals with comparable fitness values.

Once the crowding distance is obtained in the manner described above, this is then used as a tie-breaker strategy when choosing between individuals with the same Pareto rank. In other words, during selection, NSGA-II first compares the Pareto front rank; the individual with the lowest (i.e. best) rank is selected for mating. For individuals with equal Pareto front rank, the one with the higher crowding distance is selected. Once the mating phase for that generation is complete, the algorithm proceeds to the *survival* phase. For this step, NSGA-II uses a so-called "$(\mu + \lambda)$" selection strategy: once a new population of λ 'offspring' individuals has been formed via genetic operations performed on the original population of μ 'parent' individuals, the entire pool of $\mu+\lambda$ individuals is then evaluated and ranked based on their Pareto front rank and crowding distance. The topmost μ individuals from this pool are then selected to become the next generation of 'parent' individuals. In other words, by the end of the selection process, an equal number of individuals with lower Pareto front rank and higher crowding distance survive as the next population. Figure 3 presents a flowchart of the NSGA-II algorithm in the context of this work on algorithmic trading.

3 Literature review

3.1 Directional changes

Algorithmic trading has traditionally focused more on physical time data as the main algorithmic substrate in the academic literature (Long et al, 2022a; Christodoulaki et al, 2022, 2023). Nevertheless, newer literature seems to suggest that the use of directional changes has gained significant traction in recent years. The first work to introduce the concept of directional changes (DC) in financial markets was Guillaume et al (1997). The DC framework was subsequently formalised by Tsang (2010). Following this, the DC framework found wide application in the development of trading strategies within the Foreign Exchange (FX) market, due to its ability to represent irregularly spaced high-frequency tick data in an intuitive manner. For example, Aloud

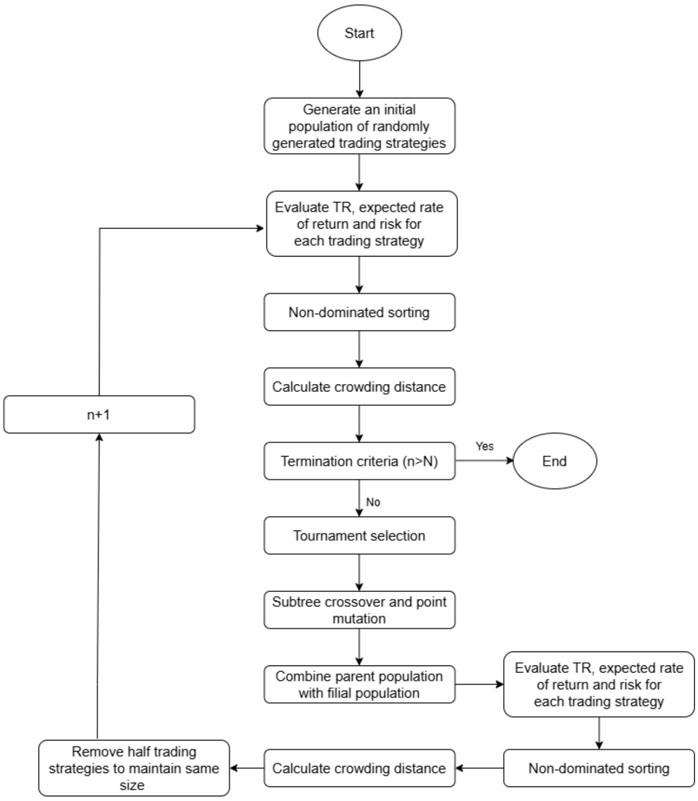

Start
Generate an initial population of randomly generated trading strategies
Evaluate TR, expected rate of return and risk for each trading strategy
Non-dominated sorting
Calculate crowding distance
n+1
Termination criteria (n>N)
Yes
End
No
Tournament selection
Subtree crossover and point mutation
Combine parent population with filial population
Evaluate TR, expected rate of return and risk for each trading strategy
Remove half trading strategies to maintain same size
Calculate crowding distance
Non-dominated sorting

Fig 3 Flowchart of the NSGA-II algorithm. n represents the current generation and N indicates the number of generations (50 in this paper)

(2012, 2015) developed trading strategies that reframed the widely used *trend following* and *contrarian* strategies under the DC framework; their work focused on how traders will react, and how adaptable such DC-based strategies are in response to FX market movements. Later, Aloud (2016a) built an automated DC trading strategy that could learn the size and direction of periodic patterns from historical asset prices. Similarly, Bakhach et al (2016a) introduced the Backlash Agent, a novel contrarian trading strategy, exhibiting strong performance within the FX market.

Additionally, Glattfelder et al (2011) discovered 12 ‘scaling laws’ that could hold under 13 foreign exchange pairs.[1] Based on this study, Aloud and Fasli (2016) proposed an additional four scaling laws under the DC framework; their work entailed an assessment of the efficacy of these scaling laws in the context of the FX market. Other work by Ao and Tsang (2019) similarly focused on using scaling laws, by exploring the relationship between DC and OS events.

More recently, machine learning has gained popularity within directional changes. Gypteau et al (2015) was the first work to use a GP-based approach to predict the future price movement of financial markets. Subsequently, Adegboye and Kampouridis (2021) and Adegboye et al (2021) applied classification and regression algorithms to predict when a trend would reverse, while Adegboye et al (2023) used a genetic algorithm to optimise the recommendations of multiple DC thresholds. Rostamian and O’Hara (2022) employed a Convolutional Neural Network Long Short-Term Memory (CNN-LSTM) model to predict DC events. Long et al (2022b) demonstrated the advantages of combining indicators derived from directional changes alongside indicators from technical analysis. Finally, Rayment and Kampouridis (2023, 2024) applied reinforcement learning to learn from directional changes in FX data.

While these promising results highlight the effectiveness of DC-based approaches in both the FX and stock markets, with or without the utilisation of machine learning algorithms, a notable gap emerges. To the best of our knowledge, all published DC works utilise single objectives, or at most, aggregate metrics that combine multiple objectives into a single metric, such as the Sharpe ratio. As previously mentioned, such aggregate metrics have the disadvantage of oversimplifying and potentially misrepresenting the relationship between the different components involved, therefore any strategies developed on the basis of such metrics could be sub-optimal. Therefore, to fill the above gap, we proposed a novel NSGA-II algorithm directly optimising TR,

[1] Scaling laws are empirical observations, similar to stylised facts in physical time frameworks, which are commonly observed in markets. Scaling laws capture statistical regularities and tendencies in financial data.

$\mathbb{E}[\mathsf{RoR}]$, and risk under the DC framework. The proposed method enables the discovery of a diverse set of Pareto-optimal solutions that balance a trade-off among the three fitness functions.

3.2 Multi-objective optimisation

Optimisation techniques are commonly applied in financial decision-making, ranging from wealth allocation strategies to algorithmic trading and multi-objective optimisation (MOO), as explored in this study (Chen et al, 2022; Leung and Wang, 2020). In particular, MOO has attracted a lot of attention in the financial literature (although not in the DC domain as of yet) and in algorithmic trading in particular. The reason behind this is that the real-world traders naturally pursue a trade-off between the conflict objectives, such as maximum excess return and minimum risk. However, most traders do not have a precise expectation of the objectives. Furthermore, Lai et al (2021) show that the low volatility effect is primarily driven by stocks with high specific risks, rather than a monotonic relationship between lower volatility and higher returns. To deal with this issue, Leung and Wang (2020) used a Collaborative Neurodynamic Optimisation approach for bi-objective portfolio selection, where they trained an ensemble of neural networks to predict asset allocation using a bi-objective function considering both return and risk. The ensemble collaborates during training to explore a diverse set of solutions, and the final set of solutions then undergoes Pareto filtering in order to obtain the final set of Pareto-optimal solutions. Similarly, Wu and Tsai (2014) applied three fuzzy goal programming models to simulate the uncertain satisfaction of trades when balancing error minimisation and excess return maximisation. The proposed model successfully identified an index tracking portfolio with a lower error while maintaining a similar excess return with the 0050 index fund, one of the most popular exchange-traded funds in Taiwan. Moreover, Wu et al (2022) proposed a three-step model to help traders determine the number of stocks in the portfolio including the stock filter. All the above studies addressed the challenge that real-world traders often struggle to find optimal solutions and introduced improvements to assist them in determining suitable solutions. Although this work does not involve portfolio management, it also motivates us to propose trader-preference scenarios to help traders select the optimal solutions from the Pareto front generated by the multi-objective optimisation algorithm.

Additionally, searching such trade-offs among the conflict objectives requires robust optimization techniques, where evolutionary algorithms (EAs) have been widely applied in algorithmic trading, particularly in multi-objective optimization tasks. Recent works in the area include de Almeida et al (2016), which proposed a differential evolution model analyzing four technical indicators, aiming to maximise profit, and minimise the level of risk and the number of trading actions. Furthermore, Kim and Enke (2016) proposed a rule change trading system using a genetic algorithm to maximise profit and payoff ratio. More recently, Atiah and Helbig (2019) conducted an in-depth study regarding the performance of three state-of-the-art dynamic multi-objective optimisation algorithms. They evaluated dynamic vector-evaluated particle swarm optimisation, a multi-objective particle swarm optimisation with crowded distance, and a dynamic non-dominated sorting genetic algorithm in the foreign exchange market. This particular work used NSGA-II, which is also the chosen MOO algorithm in our paper. In addition, Karasu et al (2020) developed a new model based on support vector regression with a wrapper-based feature-selection approach employing multi-objective optimisation. The primary aim of their work was to forecast future prices of crude oil, demonstrating the potential of this advanced methodology in predictive analysis. Noticing that feature engineering was overlooked in previous studies, Zeng et al (2023) proposed a random forest-based algorithm (I-NSGA-II-RF) that incorporates a three-stage feature engineering process. Their approach aims to identify optimal feature sets that influence stock prices, achieving higher accuracy and lower computational time compared to deep learning models.

With regards to directional changes, Long et al (2023) was the first work to introduce multi-objective optimisation within a DC framework. Our current article extends Long et al (2023) in several ways: (i) we examine a three-objective GP problem instead of a two-objective one, and introduce a corresponding three-objective variant of the Sharpe Ratio; the latter is used in two ways in this work: one, as the fitness function in the context of an SOO strategy serving as a benchmark algorithm, and two, as a post-MOO final decision strategy for selecting a single solution from the final front of Pareto-optimal solutions. (ii) we double the number of datasets we experiment with. (iii) we additionally benchmark against three standard 'zero-crossing' trading strategies based on technical analysis indicators, together with a trading strategy using a start-of-the-art deep learning model. We also benchmark against a Transformer-based architecture, due to its ability to capture complex temporal dependencies in financial data. And finally, (iv) we provide a more in-depth presentation of our methodology and results discussion.

To sum up, evolutionary algorithms have been used in conjunction with multi-objective optimisation in the domain of algorithmic trading; additionally, the incorporation of technical indicators has been proven to enhance the performance of these algorithms, as demonstrated in prior research (de Almeida et al, 2016; Kim and Enke, 2016). The present study builds upon these foundations by applying both DC-based and fixed-time technical indicators in the context of the NSGA-II framework, and thereby exploring the effectiveness of multi-objective optimisation under the DC framework. As an alternative approach to summarising the stock market into a series of significant events, DC can provide valuable insights from a new perspective while reducing market noise. Together with the traditional fixed-time technical indicators, the DC indicators could enrich the information used to train the proposed MOO3 algorithms, leading algorithms to discover more stable and profitable trading strategies.

4 Methodology

This section presents in detail the methodology followed in our work. Multi-objective optimisation (MOO) in algorithmic trading presents several challenges that arise from the inherently conflicting nature of financial objectives and the complexities of real-world markets. Traders must often balance maximising returns with minimising risk, objectives that naturally oppose each other. Additionally, financial markets are dynamic and noisy, making the identification of stable and robust trading strategies difficult. MOO algorithms must, therefore, not only deal with these conflicting goals, but also adapt to the rapidly changing market conditions. In addition, another challenge in this domain is selecting a final solution from a set of Pareto-optimal solutions. While MOO provides a diverse range of trade-offs, practitioners typically require a single, actionable strategy. Without a structured selection mechanism, choosing from the Pareto front can be subjective and inconsistent. Addressing these challenges is crucial for developing practical and effective algorithmic trading strategies. Our methodology below addresses these issues.

This study's proposed framework directly addresses these issues by integrating directional changes (DC) with genetic programming (GP) and using the NSGA-II algorithm for MOO. The role of the GP algorithm is to use evolutionary techniques, to efficiently explore the space of all possible 'programs' (i.e. trading *recipes*) that can be defined from a predefined set of *functions* and *terminals*. In this case, the GP terminals consist of DC-related indicators, as well as technical analysis (TA) indicators. The end-goal of the algorithm is to evolve trading strategies, which are optimal in the multi-objective sense; specifically, we aim to discover strategies that achieve a desirable balance between total return, expected rate of return, and risk. To this end, we employ NSGA-II (Deb et al, 2002) as the main GP evolution strategy. Once the set of suitable, Pareto-optimal solutions has been obtained via this process, we make a final, singular choice from within that set, by using a predefined criterion designed to reflect trader preference (i.e. the desired balance among the various objectives). By employing a modified Sharpe Ratio (mSR), the framework provides a systematic way to select a single strategy from the Pareto front, tailored to individual trader preferences, thus enhancing the usability and relevance of the generated solutions. The sections that follow explore the different parts of the process described above in more detail.

4.1 GP Representation

As with all evolutionary approaches, GP is a population-based approach, meaning that it requires a population of 'individuals', which evolves over a number of generations (i.e. iterations of the algorithm, see Algorithm 1) to adopt certain useful characteristics. Individuals (here corresponding to trading 'recipes') are represented as computer programs in the form of *strongly-typed syntax-trees*, where the tree nodes consist of appropriate functions and terminals.

The Function set consists of the logic operators '`AND`' and '`OR`', and the comparison operators '$>$' and '$<$'. The Terminal set consists of 28 DC indicators (listed in Table 1), 28 physical time TA indicators (listed in Table 2), and an *ephemeral random constant* (ERC), which assigns a random value between 0 and 1 each time one is inserted into a GP tree. All indicators are normalised in the range $[0, 1]$. Figure 4 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 fixed throughout the evolutionary process. The 'root' note is always an "If-Then-Else" (`ITE`) function, which decides whether to perform a 'Buy' or 'Hold' action, based on the (boolean) outcome of the leftmost branch. In this example, the first branch checks if the OSV indicator is greater than 0.22 and the 10-day N_{DC}

indicator is greater than -0.68; if both statements are true, then the leftmost branch evaluates to `true`, causing the `ITE` function to invoke a 'Buy' decision. Otherwise, if the branch evaluates to `false`, a 'Hold' decision is invoked instead. In our setup, when the GP tree outputs a 'Buy' signal, a long position is opened at the next available price. If the signal is 'Hold', no position is entered. For simplicity, we assume full capital allocation per trade, executed as a market order without slippage, latency, or partial fills. This stylised setup allows us to isolate and evaluate the effectiveness of the evolved trading rules without introducing execution complexity.

Note that 'Sell' actions are not part of the GP tree. To decide when a 'Sell' action will occur, we consider two things: the number of days n since the asset was bought, and its current price, or more specifically the percentage r by which the stock price has increased since its purchase. Specifically, our strategy for selling an asset is to sell *either* after n days have already passed from the initial purchase, *or* when a price increase of r% has occurred, whichever comes first. This provides a consistent and rule-based position control mechanism for closing trades, enabling fair comparison across evolved strategies. Note that short-selling is not allowed in this trading strategy. Whenever a trade is completed (a buy action and a corresponding sell action have occurred), 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.

4.2 Model evaluation

We evaluate each individual's performance, by considering three financial metrics, serving as the GP objectives from which suitable Pareto fronts will be obtained (i.e. for the purposes of the NSGA-II algorithm). These metrics are: total return (TR), expected rate of return ($\mathbb{E}[\mathsf{RoR}]$), and risk (Risk). The first two are maximisation objectives, and the third one is a minimisation one. Equations 2 - 4 present the relevant formulae:

Table 1: DC indicators; see also Aloud (2016b).

Indicator	Description	Periods (days)
TMV	*Total price movement*; this is defined as the price movement between the extreme point at the beginning and end of a trend, normalised by the threshold θ.	N/A
OSV	*Overshoot Value*; this 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 that the DC spends on *uptrends* compared to *downtrends* over the selected period.	10, 20, 30, 40, 50

Table 2: Physical time (technical analysis) indicators; see also Kelotra and Pandey (2020).

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

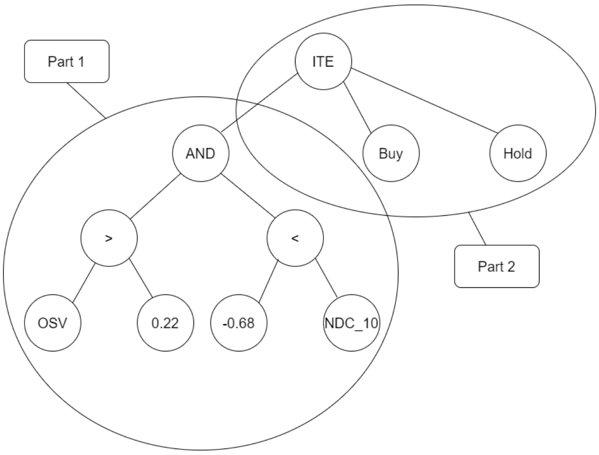


Fig 4 Sample GP tree. If OSV is greater than 0.22 and N_{DC} for 10 days is greater than -0.68, then we get a signal for a buy action; otherwise, we hold; 0.22 and -0.68 are two random values generated by ERC

$$\mathsf{RoR}(i) = \frac{(1-c)\cdot P_s(i) - (1+c)\cdot P_b(i)}{(1+c)\cdot P_b(i)} \cdot 100\% \tag{2}$$

$$\mathsf{TR} = \frac{\sum_i \Big[(1-c)\cdot P_s(i) - (1+c)\cdot P_b(i)\Big]}{(1+c)\cdot P_b(i_0)} \cdot 100\% \tag{3}$$

$$\mathsf{Risk} = \sqrt{\mathrm{Var}[\mathsf{RoR}]} \tag{4}$$

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

$P_b(i_0)$ denotes the buy price of the first ever trade event for that period, and c is the transaction cost.

The rate of return (RoR) is a measure of how profitable a particular trade turned out to be. It is a particularly useful metric for short-term investors, as it allows them to evaluate individual trades. The expectation $\mathbb{E}[\mathsf{RoR}]$ over all such events in a given period (e.g. the training or test periods) is therefore a measure of how profitable trades were for that period on average, given a particular trading strategy. The total return (TR) on the other hand, is a measure related to the overall return over a certain period, defined as the total profit over that period, expressed relative to the purchase price of the first event in the sequence. Note that its value is not necessarily capped at 100%. Its advantage over the $\mathbb{E}[\mathsf{RoR}]$ as a metric is that, in the presence of more trade events having a similar rate of return, the TR will be higher, accurately reflecting the fact that the trades during that period resulted in better overall profit. Lastly, Risk acts as an indicator of the uncertainty and potential for financial loss associated with the fluctuation of the rate of return.

For completeness, we also introduce in Equation 5 an aggregate metric, namely the Sharpe ratio (Sharpe, 1994), which is intended to be applied to entire portfolios rather than individual datasets. This is defined as:

$$\mathsf{Sharpe\ ratio} = \frac{\mathbb{E}[R - R_f]}{\sqrt{\mathrm{Var}[R]}}, \tag{5}$$

where R here reflects the expected rate of return for a given portfolio, and R_f is the risk-free rate; for the sake of simplicity, we assume here that each dataset is adequately represented by its expected rate of return, and that R_f is constant. We will use the Sharpe ratio in the results section to evaluate the risk-adjusted return of each algorithm.

4.3 Selection and Genetic operators

The GP algorithm above requires two separate selection events. The first selection event determines the 'mating pool', i.e. which individuals in the population will undergo genetic operations in order to produce new 'offspring' (i.e. candidate solutions). For this selection event, we use tournament selection as the GP selection method. To identify the winner of a tournament event, we consider the Pareto front rank and the crowding distance of each of the tournament 'contestants'. Specifically, after obtaining a number of k random individuals from the population, who are to compete in the tournament, NSGA-II first compares the Pareto front rank; the individual with the lowest rank 'wins' the tournament, and is selected for mating. For individuals with equal Pareto front rank, the one with the highest crowding distance is selected instead. The second selection event takes place after all mating operations have completed, which will have resulted in λ offspring being generated from μ parents. This selection step then determines which individuals will 'survive' to become the next generation, using a "$(\mu + \lambda)$" strategy; this effectively selects the topmost μ individuals from the resulting $\mu+\lambda$ population of parents and offspring combined, using the same fitness strategy as above (see also section 2.3).

With regard to genetic operators, we use subtree crossover and point mutation. The former exchanges subtrees between two parent individuals; the latter randomly changes a node of the resulting tree, which may be a function or a terminal, into another (compatible) function or terminal respectively.

4.4 Designating NSGA-II's final solution using a modified Sharpe Ratio

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

For the purposes of selecting such a definitive solution, we define a new aggregate metric, which effectively acts as a generalisation of the Sharpe ratio that is able to take into account total return as well as expected rate of return and risk. We include both total return and expected rate of return in our evaluation because they capture complementary aspects of strategy performance. Total return reflects the cumulative profit over the entire trading period, while expected return measures the average profitability per trade. These two metrics may diverge depending on the number and distribution of trades. For example, a strategy that achieves a high total return by capitalising on a few outlier trades may still exhibit a low expected return if the majority of trades are unprofitable. Conversely, a strategy that produces small but consistent gains on each trade can yield a high expected return, even if the total return remains modest due to the fixed trade size limiting cumulative profit.

In order to be able to calculate the metric, we first need to normalise the three objectives into the range $[0, 1]$; we do so by considering the range of values resulting for each objective in the final Pareto front. This 'modified Sharpe Ratio' (mSR) metric is then defined as per Equation 6:

$$mSR = \frac{(\widehat{\mathsf{TR}} + 1)^a \times (\widehat{\mathbb{E}[\mathsf{RoR}]} + 1)^b}{(\widehat{\mathsf{Risk}} + 1)^c} \tag{6}$$

where $\widehat{\mathsf{TR}}$ is the normalised total return value of the particular individual, $\widehat{\mathbb{E}[\mathsf{RoR}]}$ is its normalised expected rate of return, $\widehat{\mathsf{Risk}}$ is its normalised risk value, and a, b, and c are weights that determine the importance of each term in the metric, where we impose the additional constraint here that $a + b + c = 1$. Normalisation of the objectives ensures that each objective contributes equally to each term of the mSR in terms of strength. Furthermore, it is necessary to add 1 to each normalised value here, so that each term becomes monotonically non-decreasing as its corresponding weight increases. Under this scheme, each term can achieve a minimum value of 1, and a maximum value of 2. By adjusting the values of a, b, and c, we can control the emphasis placed on each metric within the GP process. For example, a term with zero weight will contribute a constant value of 1, thus having no influence on the end-score; a term assigned full weight of 1 results in that term dominating the calculation and all other terms being ignored due to the constraint. In the special case where b and c are both equal to 0.5 (and thus $a = 0$), the metric disregards total return, and effectively reduces to the square root of the Sharpe ratio (assuming normalised values). Once the mSR values are calculated for all solutions in the Pareto front, we select the one with the highest mSR as the final model produced by the algorithm. This model represents the trader's preferred trade-off among the objectives and is subsequently evaluated on the test set.

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

Table 3: International stock indices considered in this study. These represent the broader markets from which stock-level datasets were also obtained.

Dow Jones Industrial Average (DJIA)
NASDAQ Stock Market (NASDAQ)
New York Stock Exchange (NYSE)
Russell 2000 Index
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
Financial Times Stock Exchange 100 Index in the United Kingdom

5 Experimental set up

In this section, we present how our experiments are set up: we describe the data we used, the benchmark algorithms / strategies against which we compare our proposed approach, what metrics we used to evaluate performance, and how we performed hyperparameter tuning for the algorithms involved.

5.1 Data

This study uses 110 datasets derived from 10 international stock markets listed in Table 3. These datasets fall into two categories: (i) 100 stock-level datasets, comprising daily closing price data for 10 arbitrarily selected stocks from each market; and (ii) 10 index-level datasets, comprising daily closing price data for the representative stock indiex of each market (as listed in Table 3). Each dataset spans a 10-year period from 25th November 2010 to 24th November 2020, resulting in approximately 2,500 data points per dataset. The daily closing prices were then used to generate the indicators by using the process described earlier in Section 4. The aim of using a broad geographic and economic spread — including markets from North America, Europe, and Asia — is to evaluate the generalisability and robustness of the proposed algorithms across diverse market structures and volatility conditions.

The data was sourced from `Yahoo! Finance`. Our evaluation methodology involved dividing each dataset into three separate parts: the initial 60% of the data served as the training set, the subsequent 20% as the validation set and the final 20% as the testing set. During parameter tuning, the algorithms were trained using the training set and evaluated on the validation set. Once the optimal set of parameters had been determined, the GP was trained one final time using the combined training and validation set, and subsequently applied to the test set to assess the performance of the model.

5.2 Benchmarks

In this paper, we propose a novel multi-objective optimisation GP, denoted here as ‘MOO3’, designed to optimise three objectives: total return, expected rate of return, and risk. We compare our approach against the following benchmarks:

- A single-objective optimisation GP, denoted here as ‘SOO’, which uses the ‘modified Sharpe Ratio’ aggregate metric (see Equation 6) as its fitness function.
- Standard strategies relying on the straightforward application of popular technical analysis (TA) indicators, as representatives of a physical time setup: a zero-crossing strategy based on the moving average convergence divergence (MACD) indicator (Vaidya, 2020), a zero-crossing strategy of long- and short-term moving averages of the on-balance-volume (OBV) indicator (Deprez and Frömmel, 2024), and a threshold-crossing strategy based on the momentum (MTM) indicator (Fong et al, 2012).
- A single-objective Transformer-based model (Vaswani et al, 2017) due to its ability to capture long-range dependencies and complex patterns in financial data. We use the `TransformerModel` from the `Darts` Python library for financial time series forecasting. Like all other SOO algorithms in this article, the Transformer uses the Sharpe ratio as its objective. We performed hyperparameter

tuning through a grid search process as per Gal and Ghahramani (2016) and Gao et al (2021); the final model was configured with 4 encoder layers, 4 attention heads, a dropout rate of 0.1, and a batch size of 32. It was trained for 50 epochs using an Adam optimiser with a learning rate of 5e-4. The input and output chunk lengths were set to 30 and 7, respectively. Lastly, we used the Bernoulli likelihood function to simulate a classification process. The above Transformer is using the MOO's trading strategy, as described earlier in Section 4.

- The passive trading strategy of buy-and-hold, which is a popular benchmark against active trading strategies, such as the GPs utilised in this paper.

5.3 Trader-preference scenarios

As mentioned in section 4.4, while a multi-objective optimisation framework is necessary and beneficial to obtain solutions that optimise conflicting objectives, in the end a trader will require a single trading strategy to deploy. Therefore it is imperative to be able to designate a single solution from the Pareto front, in a manner that allows the trader to specify their preference over the various objectives. We do this here via the modified Sharpe Ratio, as introduced in Equation 6, by modifying the different weight values (a, b, c), where a corresponds to the weight given to total return, b to the expected rate of return, and c to risk. In our experiments, we have focused on seven different scenarios/setups:

[a=0.5, b=0.5, c=0] : The final solution focuses equally on total return and expected rate of return.

[a=0, b=0.5, c=0.5] : The final solution focuses equally on expected rate of return and risk.

[a=0.5, b=0, c=0.5] : The final solution focuses equally on total return and risk.

[a=0.33, b=0.33, c=0.33] : The final solution focuses equally on all three metrics.

[a=1, b=0, c=0] : The final solution focuses only on total return.

[$\mathsf{a}=0,\ \mathsf{b}=1,\ \mathsf{c}=0$] : The final solution focuses only on expected rate of return.

[$\mathsf{a}=0,\ \mathsf{b}=0,\ \mathsf{c}=1$] : The final solution focuses only on risk.

These setups allow us to consider different extreme cases, where only one or two metrics are being considered, and also the case where all three metrics are equally being considered.

5.4 Parameter tuning

The parameter tuning process consists of two stages. The first stage involves optimising the GP algorithm's standard parameters, which are then used identically over all datasets. These parameters are the maximum depth, which controls the depth of the GP trees; the population size, which controls the number of individuals (trading strategies) in the population; crossover probability, which determines when crossover and mutation will take place ($Prob_{xover} = 1 - Prob_{mutation}$); tournament size, which controls the selection pressure of the GP algorithm; and generations number, which determines how many generations a single GP run will run over.[2] To determine these parameters, we conducted a grid search over the parameters, and selected the parameter set which resulted in the best 'performance' over the validation dataset; in this context, 'performance' was evaluated specifically as the Sharpe Ratio over the entire portfolio. For computational reasons, rather than use all datasets, the above tuning was only performed using 10 datasets, chosen at random. The above optimal GP parameter set is then used universally across all datasets and scenarios, to train the set of final GPs (i.e. specific to each dataset and scenario) on the combined training+validation sets; this is the case for both the MOO3 approach and the SOO benchmark approach. The GP parameters and their corresponding values after tuning are summarised in Table 4.

[2] Generally, in evolutionary algorithms such as the NSGA-II, larger populations tend to improve solution diversity and prevent premature convergence. On the other hand, they increase computational cost. Similar observations can be made to other parameters such as crossover and mutation. For example, with regards to mutation, higher rates encourage greater exploration, but delay convergence. Hence there's a trade-off when selecting values for these parameters. Our thorough parameter tuning approach described above is to ensure that we use values that take into account this trader-off.

Table 4: Selected parameters of the GP algorithm after parameter tuning.

Parameter	Value
Max depth	6
Population size	500
Crossover probability	0.95
Tournament size	2
Number of generations	50

The second stage of parameter tuning involves optimising the remaining three parameters for our problem, on a per-dataset-and-scenario basis. The first two parameters are related to the trading strategy used by the GP. As previously mentioned, a 'sell' action is triggered automatically if n days have passed since the 'buy' event, or if the stock price has increased by $r\%$, whichever comes first. Here, the variables r and n signify the anticipated future price movement and a temporal constraint respectively, and need to be tuned accordingly for each dataset/scenario pair. The third parameter is the threshold θ, which controls the definition of the significant event under the DC framework. Similar to r and n, instead of optimising θ universally across all datasets and scenarios as we did for the parameters of the GP-based algorithms, we opt instead for tailored values for each dataset/scenario pair. For the above purposes, we considered 3 representative values for n, 4 values for r, and 5 values for θ, leading to a grid of $3 \times 4 \times 5 = 60$ configurations per dataset/scenario pair. The configuration space for these three parameters is presented in Table 5.

Armed with the above, we conducted tuning experiments as follows: we conducted 5 experiments per parameter configuration, for each dataset/scenario pair. Each such experiment resulted in a population of solutions (i.e. trading strategies), with a corresponding Pareto Front; the modified Sharpe Ratio was thus used to select a definitive trading strategy per experiment, and the performance of each such definitive trading strategy was then evaluated on the validation set. The resulting overall performance of each configuration was then obtained as the average performance over

Table 5: Configuration space for the trading strategy's n days and $r\%$, and DC's θ.

Parameter	Configuration space
n (days-ahead of prediction)	1, 5, 15
r (percentage of price movement)	1%, 5%, 10%, 20%
θ (threshold of DC)	0.001, 0.002, 0.005, 0.01, 0.02

the 5 corresponding runs for that configuration, and the configuration with the best average performance was selected for each data/scenario pair. Note that, unlike the GP parameters, the above process was conducted separately for the MOO3 and SOO approaches. In other words, for each of the algorithms, the number of tuning experiments performed per scenario was: 110 datasets × 60 configurations × 5 repetitions = $33,000$ experiments.

Note that, to some extent, the choice of optimal GP parameters depends on r, n, and θ, and vice-versa, in a cyclical manner. To work around this cyclical dependency, we bootstrap this process as follows: first, we use an initial parameterisation for the GP serving as a reasonable 'prior' (in our case, we used values based on previous work); this allows performing a first round of tuning over r, n, and θ for all dataset/scenario combinations, as above. With these resulting 'prior' values for r, n, and θ at hand, we then proceed with the first stage of the tuning process, as described above.

6 Results and analysis

In this section, we report and analyse the performance of the multi-objective optimisation GP approach (MOO3) against the benchmark approaches. We start this section by visualising the performance of MOO3 against corresponding single-objective approaches (SOO) for each of the 7 scenarios presented in section 5.3. We then perform a statistical analysis, comparing single and multi-objective optimisation GP algorithms, followed by a comparison against the more traditional, TA-indicator based benchmarks, Transformer, and finally against the buy-and-hold strategy.

In order to compare the MOO3 approach statistically against each of the benchmarks, we perform multiple experiments as follows: first, we run the MOO3 algorithm 50 times for each of the 110 datasets, and each of the 7 scenarios. This means we obtain 50 different, final Pareto fronts per dataset and scenario. For each of these runs, we obtain the best model from the training set with respect to the corresponding modified Sharpe Ratio for that scenario. Given that the proposed algorithm is designed as a maximisation problem, the model exhibiting the highest modified Sharpe Ratio is selected from the Pareto-optimal set. We then evaluate this model on the test set, and measure its performance using a variety of evaluation metrics. Finally, for each evaluation metric, we obtain the average such test-set performance over the 50 runs. Similarly, for the SOO algorithm, we run the SOO algorithm 50 times per setup and dataset, obtain the fittest solution arising from the training-set in each case, and apply it to the test-set; we then average this test-set performance over all 50 runs. The end-result is that we have an average performance from MOO3, and an average performance from SOO to compare, per dataset and scenario. This process allows us to make a direct comparison of the SOO and MOO3 models, since they have both been selected using the same aggregate metric and 'trader-preference' weightings in each case. Finally, we also obtain the results of the TA-indicator based, Transformer-based, and buy-and-hold benchmark trading approaches, as applied directly onto the test set.

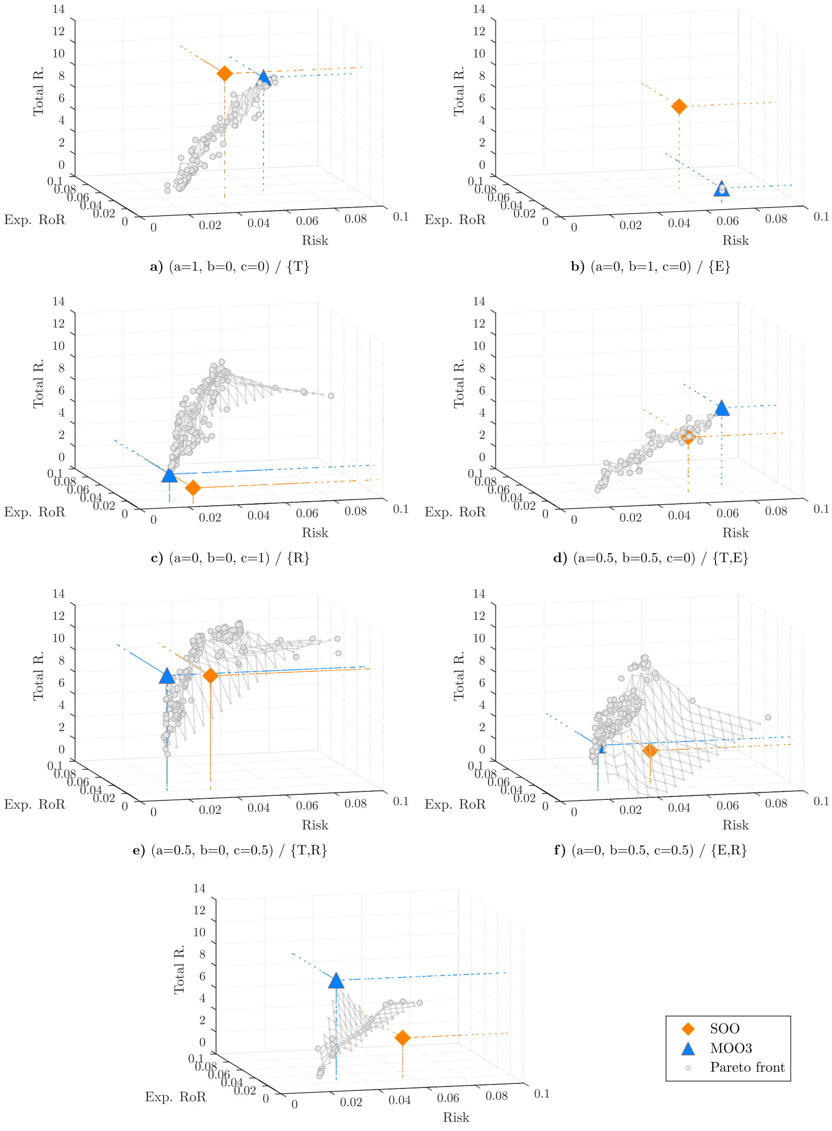
6.1 Pareto front

We begin by presenting the results of the MOO3 algorithm, and examining the placement of the MOO3-designated solutions (and the Pareto front solutions more generally) compared to the single-objective optimisation (SOO) solutions, for each of the seven scenarios considered. For illustration, we use a single GP run for each scenario, focusing on the Apple stock. Figure 5 displays the generated solutions and resulting interpolated surface making up the Pareto front (grey mesh/markers), the designated MOO3-solution from that front (blue triangle marker), and the corresponding SOO solution (orange diamond marker), for each of the seven scenarios. The MOO3 and SOO solutions are accompanied by 2D projections to aid interpretation and comparison. Each scenario is identified via its $[a, b, c]$ weight-triplet; to enhance clarity, each triplet is also expressed as the corresponding subset of the set {T,E,R}, where the choice out of T, E, and/or R represents how preference was distributed among total return, expected rate of return, and risk respectively in the weighting; e.g. {T,R} indicates the scenario where total return and risk are valued equally and exclusively, therefore corresponding to the weighting $[a{=}0.5,\ b{=}0,\ c{=}0.5]$.

Looking at the figure, we observe the following:

- In the {T} and {T,E} scenarios, the SOO solution dominates a handful of solutions of MOO3's Pareto front, but neither SOO nor the MOO3 designated solution dominate each other.
- In the {E} scenario, SOO dominates the MOO3-designated solution, as well as the (unusually small) Pareto front.
- In the {R}, {T,R}, {E,R}, and {T,E,R} scenarios, the MOO3 designated solution dominates the SOO solution, and SOO does not dominate any solutions on the Pareto front.

The above suggests that, in this example at least, MOO3 had superior performance when non-singular objectives were considered, showing its ability to consider multiple objectives in a natural manner, driven by the concept of the Pareto front, as compared to an aggregate approach which imposes an arbitrary relationship between the objectives. By contrast, when dealing with single-objective scenarios, SOO was able to compete well with MOO3; in particular, when considering single-objective scenarios, SOO achieved better total return and expected rate of return in the {T} and {E} scenarios respectively. However, MOO3 achieved better risk for the {R} scenario; this



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

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

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

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

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

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

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

Fig 5 Representative run of the MOO3 and SOO algorithms for the Apple stock. The SOO and MOO3-designated solutions are overlaid onto the MOO3 Pareto front, for all seven scenarios considered. Risk, expected rate of return, and total return are shown here in their original units, rather than the normalised ones used during fitness evaluation

demonstrates that MOO3 still has the ability to obtain good-quality single-objective designated solutions that may compete with explicitly single-objective approaches, even though the Pareto front driven evolution process has to take into account and accommodate multiple objectives, partly acting as constraints. In the next section, we perform statistical analyses over multiple runs, in order to examine how the two algorithms compare over multiple runs and datasets more generally.

6.2 Convergence Analysis

To better understand the optimisation behaviour of the proposed GP algorithms, we include a convergence analysis, which reveals how performance evolves across generations and whether the search consistently leads to improved solutions. Specifically, we record the best individual (SOO) and Pareto front (MOO3) fitness at Generations 1, 10, 25, and 50 in the training set of a single GP run. For each generation, we analyse how the objective metrics — total return (TR), expected rate of return ($\mathbb{E}[\mathsf{RoR}]$), risk (Risk) — evolve during the evolutionary process. Due to space constraints and the large number of datasets used in our study (110 in total), we present the results for two representative stock datasets for each of the SOO and MOO3 on the basis of an $[a=1,\ b=0,\ c=0]$ weighting. These stocks were selected to illustrate typical convergence behaviour observed across the broader dataset collection.

Figure 6 displays the performance of individuals in the population at each selected generation for the chosen SOO and MOO3 runs, respectively. The plots clearly demonstrate a progressive improvement in performance across generations. For the SOO, Figure 6a and Figure 6b present the best individual in the population across the generations. The value of TR for both stocks increases rapidly between Generations 1 and 10, after which the rate of improvement slows considerably. This pattern indicates that the majority of optimisation occurs early in the evolutionary process, with subsequent generations yielding only minor improvement, providing clear evidence of convergence in the SOO evolutionary process.

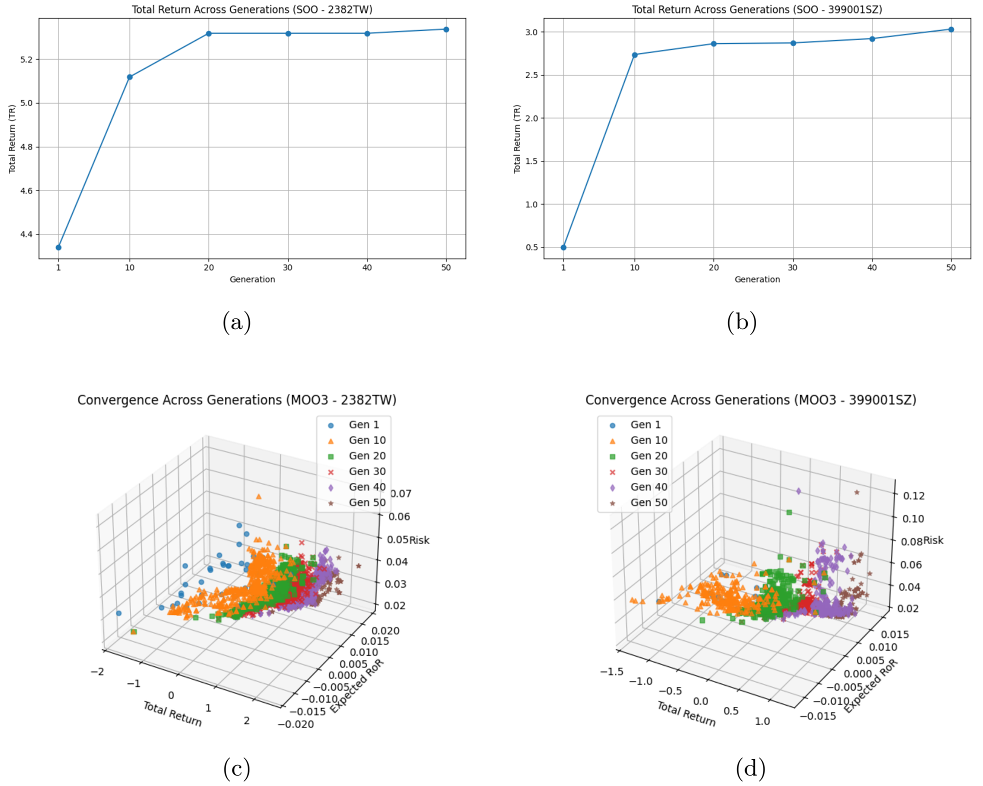
In contrast, the MOO3 results, illustrated in Figures 6c and 6d, demonstrate continuous improvement in the trade-offs among the three objectives, reflected by a progressively refined Pareto front. Both Figures show rapid improvements in the metric values of the Pareto Front (particularly Total Return, as it is given 100% weight), followed by only minor improvements in later generations, and convergence around generation 40-50.

In summary, the convergence analysis shows that both the SOO and MOO3 algorithms improve steadily over generations and eventually reach stable solutions. Most of the progress happens early on, with smaller improvements later, which suggests that the algorithms are working efficiently. For MOO3, the gradual shaping of the Pareto front across generations shows that the algorithm is learning to balance the different objectives well.

6.3 Comparison between multi-objective and corresponding single-objective approaches

Table 6 presents the summary statistics for SOO and MOO3 across five metrics: total return (TR), expected rate of return ($\mathbb{E}[\mathsf{RoR}]$), risk (Risk), and the standard Sharpe ratio (SR) for the whole portfolio. The portfolio Sharpe ratio assumes an equal distribution of the 110 stocks (i.e. each stock has the same weight in the portfolio)[3]; since this figure is calculated over the whole portfolio of stocks, there's no summary statistics to calculate, just a single value, which is presented in the table. For the remaining four metrics (TR, $\mathbb{E}[\mathsf{RoR}]$, Risk) we present the mean, median, standard deviation, maximum and minimum values across the results of the 110 stocks. The left hand-side of the table presents the results for the single-objective (SOO) GP,

[3] This assumption is made for simplification purposes. It is beyond the scope of this work to look into portfolio optimisation algorithms, which would seek to find the optimal set of portfolio weights for all stocks present.



(a)

(b)

(c)

(d)

Fig 6 Convergence behaviour of the SOO and MOO3 algorithms across generations. The plots show how the fitness function evolves over time for two representative stock datasets (2382TW and 399001SZ) under each optimisation framework. Results are recorded at generations 1, 10, 20, 30, 40, and 50 to illustrate the improvement of the population as evolution progresses.

while the right hand-side presents the results for the multi-objective MOO3 algorithm. Results are separated by the seven different sets for Equation 6's $[a, b, c]$ weights. To enhance clarity, similarly to Figure 5, we also present the algorithms in the format $\mathsf{SOO}_{\{.\}}$ (or $\mathsf{MOO}_{\{.\}}$), with the denominator able to represent any subset of $\{\mathsf{T},\mathsf{E},\mathsf{R}\}$, as explained in Section 6.1 (e.g. $\mathsf{MOO}_{\{\mathsf{T},\,\mathsf{E}\}}$ denotes a MOO3 algorithm where the designated solution was selected on the basis of an $[a{=}0.5,\ b{=}0.5,\ c{=}0]$ weighting).

As we can observe, MOO3 generally has higher mean and median TR across different weight setups compared to SOO. In addition, MOO3 shows a higher mean and median $\mathbb{E}[\mathsf{RoR}]$ compared to SOO for most weight setups. With regards to Risk, both algorithms show similar risk levels, but MOO3 offers small improvements in a few weight setups. Lastly, MOO3 has a higher Sharpe Ratio in six out of the seven weights setups, suggesting better risk-adjusted portfolio returns.

We perform a Kolmogorov-Smirnov (KS) test for each of the pairwise comparisons between SOO and MOO3 on three metrics: Total Return (TR), Expected Rate of Return ($\mathbb{E}[\mathsf{RoR}]$), and Risk. The null hypothesis for each KS test is that the two samples (SOO and MOO3) come from the same population distribution. Since we are conducting multiple comparisons, we apply the Holm-Bonferroni correction to control the family-wise error rate at a significance level of 5%. Specifically, we adjust the significance threshold required for each of the three comparisons (TR, $\mathbb{E}[\mathsf{RoR}]$, Risk) within each $[a, b, c]$ weights setup. This is done by arranging p-values from smallest to largest, and assigning a corresponding rank from 1 to 3 respectively. The minimum acceptable p-value for a comparison to be considered statistically significant is then decided based on its rank. The adjusted significance level for each rank is calculated as

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

where $alpha = 0.05$ for the 5% significance level, and $m = 3$, since we have 3 multiple comparisons per weight setup.

Given the three possible different ranks, we have the following adjusted significance levels :

- The smallest p-value (rank 1) must be less than $\frac{0.05}{3}$ or approximately 0.0167.
- The second smallest p-value (rank 2) must be less than $\frac{0.05}{2}$ or 0.025.
- The largest p-value (rank 3) must be less than 0.05.

Therefore, we compare the ranked p-values against these thresholds to determine if there is a significant difference between the SOO and MOO3 samples for each metric.

Table 6: Summary statistics over assets, contrasted between the SOO and MOO3 approaches for the different trader-preference scenarios. Best value for each respective metric shown in boldface. Keys: TR: total return, $\mathbb{E}$[RoR]: expected rate of return, SR: Sharpe ratio, SOO: single-objective optimisation GP, MOO3: three-objective optimisation GP. The values of a, b, c correspond to the weights as defined in Equation 6.

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

The ranked p-values indicate a significant difference if the smallest p-value is less than 0.0167, the second smallest is less than 0.025, and the largest is less than 0.05.

Table 7 presents the KS test results, along with the p-value and the adjusted significance level, across the different weight setups ($[a, b, c]$). As we can observe, the differences between MOO3 and SOO are statistically significant at the 5% level in 9 out of the 21. Particularly, MOO3 shows statistically significant improvements in TR and $\mathbb{E}$[RoR] in most weight configurations and also in managing risk in some setups. The above findings lead us to conclude that our multi-objective optimisation GP algorithm provides a more robust and efficient optimisation strategy compared to the single-objective optimisation GP, particularly in balancing return and risk.

Table 7: Kolmogorov-Smirnov test results p-values for different weight setups ($[a, b, c]$). Statistically significant results at the 5% level are denoted in bold face. Keys: TR: total return, $\mathbb{E}$[RoR]: expected rate of return, SR: Sharpe ratio.

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

While the above KS test was useful for making pairwise comparisons between the SOO and MOO3 algorithms under different weight setups and metrics, we are also interested in gaining a better understanding of the overall performance of all SOO and MOO3 algorithms across the different weight setups. The non-parametric Friedman test is particularly useful for comparing multiple algorithms across multiple datasets, as it ranks the algorithms for each metric, providing an average rank for each algorithm. A lower average rank indicates better performance. In each Friedman test (one per TR, $\mathbb{E}[\mathsf{RoR}]$, and Risk metric), the algorithms included are the SOO and MOO3 algorithms under the different weight setups. Additionally, we applied the Hommel post-hoc test to ascertain the significance of the differences between the average ranks (Demšar, 2006; Garcia and Herrera, 2008). We present both in Table 8. For each algorithm, the table shows the average rank (first column), and the adjusted $p-$value of the statistical test when that algorithm's average rank is compared to the average rank of the algorithm with the best rank (control algorithm) according to Hommel's post-hoc test (second column). When statistically significant differences between the average ranks of an algorithm and the control algorithm at the 5% level ($p \leq 0.05$) are observed, the relevant average rank is put in bold face.

As we can observe from the total return results of Table 8a, the control algorithm is $\mathsf{MOO}_{\{\mathsf{T}\}}$, i.e. the multi-objective optimisation GP algorithm that extracts the best model from the Pareto front based on the highest total return value. In addition, it statistically and significantly outperforms all other algorithms at the 5% level. Given that we are currently considering total return results in this subtable, it is unsurprising that the control algorithm is one that prioritises total return. It is also worth noting that the second and third ranked algorithms are also MOO ones, namely $\mathsf{MOO}_{\{\mathsf{T,E}\}}$ and $\mathsf{MOO}_{\{\mathsf{T,E,R}\}}$. The former gives equal weight to total return and risk ($a=b=0.5,\ c=0$), and the latter assigns equal weight to all three metrics ($a = b = c = 0.33$). This is an important finding, because it demonstrates that these multi-objective algorithms are able to outrank all SOO algorithms; hence, confirming the fact that considering multiple objectives under a MOO framework such as NSGA-II has significant benefits over an SOO algorithm which uses the aggregate fitness function directly. Similar findings can be observed for the expected rate of return and risk results (Tables 8b

Table 8: Statistical test results for average TR, $\mathbb{E}$[RoR], and Risk, according to the non-parametric Friedman test with the Hommel post-hoc test of different MOO and SOO algorithms. The subscript for each algorithm denotes which metrics were optimised. When more than one metric is present, equal weights have been assigned to each metric. Significant differences at the $\alpha = 5\%$ level between the control algorithm (appearing as the top row in each case) and the remaining algorithms are shown in boldface, indicating that the adjusted p-value is lower than 0.05.

(a) Total Return

	Avg Rank	Adj p_{Homm}
MOO$_{\{T\}}$	4.02	-
MOO$_{\{T,E\}}$	**5.60**	2.58E-03
MOO$_{\{T,E,R\}}$	**5.86**	6.34E-04
SOO$_{\{T,E,R\}}$	**5.90**	4.95E-04
SOO$_{\{T\}}$	**6.03**	2.21E-04
SOO$_{\{T,R\}}$	**6.14**	1.07E-04
MOO$_{\{E\}}$	**6.47**	1.06E-05
SOO$_{\{T,E\}}$	**6.72**	1.51E-06
MOO$_{\{T,R\}}$	**7.15**	3.28E-08
MOO$_{\{E,R\}}$	**8.32**	5.77E-14
SOO$_{\{E\}}$	**9.30**	9.98E-20
MOO$_{\{R\}}$	1**0.45**	8.41E-28
SOO$_{\{E,R\}}$	**11.01**	2.98E-32
SOO$_{\{R\}}$	**11.95**	1.95E-40

(b) Expected Rate of Return

	Avg Rank	Adj p_{Homm}
MOO$_{\{E\}}$	4.63	-
MOO$_{\{T,E\}}$	4.74	2.34E-01
MOO$_{\{T\}}$	4.81	2.17E-01
MOO$_{\{T,E,R\}}$	**5.70**	3.42E-05
SOO$_{\{T\}}$	**6.76**	8.05E-05
SOO$_{\{T,E,R\}}$	**6.89**	3.42E-05
SOO$_{\{T,R\}}$	**7.05**	1.18E-05
SOO$_{\{T,E\}}$	**7.16**	5.09E-06
MOO$_{\{E,R\}}$	**7.44**	5.21E-07
SOO$_{\{E\}}$	**8.00**	2.91E-09
MOO$_{\{T,R\}}$	**8.32**	1.06E-10
MOO$_{\{R\}}$	**10.39**	4.74E-23
SOO$_{\{E,R\}}$	**10.78**	8.44E-26
SOO$_{\{R\}}$	**12.21**	3.31E-37

(c) Risk

	Avg Rank	Adj p_{Homm}
MOO$_{\{R\}}$	2.16	-
MOO$_{\{T,R\}}$	**4.85**	1.01E-06
SOO$_{\{R\}}$	**5.11**	1.04E-07
SOO$_{\{E,R\}}$	**5.16**	6.55E-08
MOO$_{\{E,R\}}$	**5.30**	1.66E-08
MOO$_{\{T,E,R\}}$	**7.51**	2.10E-20
MOO$_{\{T\}}$	**7.68**	1.42E-21
SOO$_{\{T,R\}}$	**7.99**	8.88E-24
SOO$_{\{T\}}$	**8.30**	4.43E-26
SOO$_{\{T,E,R\}}$	**9.26**	1.03E-33
MOO$_{\{E\}}$	**9.71**	1.35E-37
SOO$_{\{T,E\}}$	**10.03**	1.82E-40
MOO$_{\{T,E\}}$	**10.15**	1.69E-41
SOO$_{\{E\}}$	**11.75**	1.52E-57

and 8c). More specifically, when looking into the expected rate of return results, the best rank algorithm is $\mathsf{MOO}_{\{\mathsf{E}\}}$, which optimises only the expected rate of return. In the next three positions there are again multi-objective optimisation algorithms that optimise different combinations of the metrics, even risk ($\mathsf{MOO}_{\{\mathsf{T,E,R}\}}$). This again shows the advantages of multi-objective optimisation. Furthermore, when looking into the risk results, $\mathsf{MOO}_{\{\mathsf{R}\}}$ ranks first, followed by $\mathsf{MOO}_{\{\mathsf{T,R}\}}$. So we again see the same pattern, where the algorithm is specifically optimised for a particular metric (Risk in this instance) tends to perform best in this specific metric, and then being followed by another multi-objective optimisation algorithm that considers two or three objectives. Lastly, it is also worth noting that in the top 2 positions of each subtable we have a $\mathsf{MOO}_{\{\mathsf{T}\}}$ variant, i.e. a MOO3 algorithm that has used total return (alone or in consideration with another metric) to extract the best model from the Pareto front. This suggests that prioritising total return in the optimisation process can lead to favourable outcomes.

To conclude, the Friedman test results highlight the effectiveness of the MOO3 algorithms over their SOO counterparts. Combining up to two metrics tends to be beneficial (or at least have comparable performance to the MOO algorithms optimising a single metric), as we have seen in the case of $\mathsf{MOO}_{\{\mathsf{T,E}\}}$, which performs well for both total return and expected rate of return. However, the trade-offs become apparent when more metrics are combined (e.g. $\mathsf{MOO}_{\{\mathsf{T,E,R}\}}$), where balancing multiple objectives leads to a slight drop in the performance of the individual metrics.

6.3.1 Summary of MOO vs SOO comparison

In this section, we conducted an in-depth analysis of the kind of improved performance that a three-objective GP algorithm (MOO3) can achieve over a single-objective GP (SOO). The algorithms were evaluated across various weight configurations for three objectives: total return, expected rate of return, and risk. The results showed that MOO3 generally outperformed SOO in terms of mean and median total return, expected rate of return, and portfolio Sharpe ratio across different weight setups. Kolmogorov-Smirnov tests confirmed that the differences between MOO3 and SOO were statistically significant in several cases. Furthermore, the non-parametric Friedman test with Hommel's post-hoc analysis revealed that MOO3 algorithms that focused on individual objectives (e.g. total return or expected rate of return) performed the best for their respective objectives, followed by MOO3 algorithms that combined multiple objectives.

Of particular note is the perhaps somewhat surprising fact that MOO3 variants consistently outperformed SOO algorithms, even in scenarios focused on single objectives. One might have expected the reverse to happen, under the understanding that the SOO algorithms are explicitly trying to optimise the single objective directly, whereas MOO3 has to take trade-offs between multiple objectives into account during the training process, as guided by the Pareto Rank instead. However, our findings suggest that the multi-objective approach proves advantageous, and that accounting for the interplay between conflicting objectives during training leads to superior trading strategies, even when optimizing for singular objectives.

6.4 MOO algorithms under different market conditions

Financial trading strategies must adapt to the different market conditions and liquidity constraints (Sun et al, 2019). Therefore, having established that the MOO algorithms can outperform the SOO algorithm, we shift our focus to understanding the MOO's performance under different market conditions. As previously discussed in Section 5, we have obtained data from 10 different financial markets: DJIA, NASDAQ, NYSE, Russel 2000, S&P500, NIFTY 50, TSEC, DAX, Nikkei 225, and FTSE 100. We are interested in understanding if the same algorithms perform consistently well under the same markets or if their performance varies, depending on how the market conditions change.

From Table 9, we conclude that $\mathsf{MOO}_{\{\mathsf{T}\}}$ consistently achieved the highest average TR values among the seven MOO3 algorithms, except in the NASDAQ market, where $\mathsf{MOO}_{\{\mathsf{E}\}}$ obtained 104.15% average TR, followed by $\mathsf{MOO}_{\{\mathsf{T}\}}$ with 97.16%. Furthermore, in terms of $\mathbb{E}[\mathsf{RoR}]$, $\mathsf{MOO}_{\{\mathsf{E}\}}$ had the highest average $\mathbb{E}[\mathsf{RoR}]$ in four out of ten markets, while $\mathsf{MOO}_{\{\mathsf{T}\}}$ achieved this in three out of ten markets. However, as a trade-off, the model which achieved the highest average TR or $\mathbb{E}[\mathsf{RoR}]$ always comes with the worst corresponding standard deviation values. In terms of Risk, $\mathsf{MOO}_{\{\mathsf{E}\}}$ had the lowest risk in nine out of ten markets. Moreover, it also could maintain the best standard deviation risk value in eight out of ten markets.

Similar to what we observed earlier, each algorithm that achieved the best average value was optimized for the respective objective. Across the ten markets, the performance of the MOO3 algorithms remains consistent, with $\mathsf{MOO}_{\{\mathsf{T}\}}$, $\mathsf{MOO}_{\{\mathsf{E}\}}$, and $\mathsf{MOO}_{\{\mathsf{R}\}}$ always ranking first in terms of their corresponding objectives. We conclude that the different markets did not affect the performance of the MOO3 algorithms.

The above findings are further supported by the Friedman test presented in Table 10. To improve clarity, the order of algorithms in the table corresponds to their ranking on the Friedman test with the values in parentheses representing the adjusted p-values. From Table 10, we can confirm that in most cases the highest ranking algorithms are then ones that were optimized for the respective objective.

6.5 Comparison of MOO algorithms to three standard TA-indicator based strategies

The previous section of this paper established that our proposed multi-objective GP algorithm is able to outperform its SOO counterpart. In the current section, we are interested in benchmarking the MOO3 algorithm against the three simple, widely-used TA-indicator based trading strategies identified in section 5.2, namely: a zero-crossing strategy for the Moving Average Convergence Divergence (MACD) indicator, which

Table 9: Average (standard deviation) results of the MOO3 approaches for the different trader-preference scenarios across various markets. Best value for each respective metric shown in boldface. Keys: TR: total return, $\mathbb{E}[\mathsf{RoR}]$: expected rate of return, SR: Sharpe ratio.

(a) NYSE

	TR	$\mathbb{E}[\mathsf{RoR}]$	Risk	SR
MOO$_{\{T,E\}}$	28.37%(0.38)	1.33%(0.02)	0.09(0.04)	0.75
MOO$_{\{E,R\}}$	15.16%(**0.12**)	1.08%(0.01)	0.06(**0.02**)	**1.78**
MOO$_{\{T,R\}}$	22.36%(0.33)	1.09%(0.01)	0.07(0.03)	0.94
MOO$_{\{T,E,R\}}$	25.36%(0.32)	1.48%(0.01)	0.07(0.03)	1.11
MOO$_{\{T\}}$	**33.02**%(0.43)	**1.65**%(0.02)	0.09(0.05)	0.68
MOO$_{\{E\}}$	18.43%(0.26)	1.16%(0.02)	0.09(0.04)	0.50
MOO$_{\{R\}}$	4.04%(0.15)	0.33%(0.01)	**0.04**(0.03)	0.40

(b) NASDAQ

	TR	$\mathbb{E}[\mathsf{RoR}]$	Risk	SR
MOO$_{\{T,E\}}$	77.97%(0.96)	3.62%(0.04)	0.16(0.09)	0.90
MOO$_{\{E,R\}}$	56.03%(0.89)	2.04%(0.03)	**0.12**(**0.07**)	0.81
MOO$_{\{T,R\}}$	57.22%(0.81)	1.96%(**0.02**)	**0.12**(**0.07**)	0.87
MOO$_{\{T,E,R\}}$	77.48%(0.96)	2.78%(0.03)	0.14(0.08)	**1.06**
MOO$_{\{T\}}$	97.16%(1.01)	3.63%(0.04)	0.16(0.11)	0.95
MOO$_{\{E\}}$	**104.15**%(1.62)	**5.21**%(0.07)	0.19(0.13)	0.79
MOO$_{\{R\}}$	31.33%(**0.55**)	2.45%(0.04)	0.13(0.12)	0.67

(c) TSEC

	TR	$\mathbb{E}[\mathsf{RoR}]$	Risk	SR
MOO$_{\{T,E\}}$	23.98%(0.26)	2.70%(0.04)	0.05(0.05)	0.75
MOO$_{\{E,R\}}$	11.78%(0.17)	1.23%(0.01)	0.03(0.02)	**0.87**
MOO$_{\{T,R\}}$	19.65%(0.31)	1.10%(0.01)	0.03(0.02)	0.85
MOO$_{\{T,E,R\}}$	19.72%(0.27)	1.96%(0.03)	0.04(0.03)	0.75
MOO$_{\{T\}}$	**24.54**%(0.28)	**5.20**%(0.12)	0.04(0.02)	0.42
MOO$_{\{E\}}$	19.56%(0.20)	3.60%(0.07)	0.03(0.02)	0.51
MOO$_{\{R\}}$	10.95%(**0.15**)	0.48%(0.01)	**0.02**(**0.01**)	0.46

(d) DAX

	TR	$\mathbb{E}[\mathsf{RoR}]$	Risk	SR
MOO$_{\{T,E\}}$	48.38%(0.33)	3.42%(0.02)	0.07(0.02)	1.62
MOO$_{\{E,R\}}$	22.59%(0.14)	2.50%(0.02)	0.05(0.01)	1.60
MOO$_{\{T,R\}}$	34.19%(0.27)	2.14%(0.01)	0.05(0.01)	1.75
MOO$_{\{T,E,R\}}$	38.99%(0.29)	2.98%(0.02)	0.06(0.02)	1.49
MOO$_{\{T\}}$	**55.39**%(0.38)	**3.94**%(0.02)	0.06(0.01)	2.34
MOO$_{\{E\}}$	40.93%(0.21)	3.79%(0.01)	0.07(0.02)	**3.16**
MOO$_{\{R\}}$	12.31$(**0.10**)	1.33$(0.01)	**0.03**(0.01)	1.86

(e) RUSSELL 2000

	TR	$\mathbb{E}[\mathsf{RoR}]$	Risk	SR
MOO$_{\{T,E\}}$	47.67%(0.70)	2.23%(0.02)	0.11(0.05)	1.30
MOO$_{\{E,R\}}$	32.35%(0.49)	1.57%(0.02)	0.07(0.05)	0.95
MOO$_{\{T,R\}}$	40.15%(0.67)	1.54%(**0.01**)	0.08(**0.04**)	1.06
MOO$_{\{T,E,R\}}$	51.26%(0.74)	2.09%(0.02)	0.09(0.05)	1.08
MOO$_{\{T\}}$	**55.72**%(0.74)	2.48%(0.02)	0.10(0.05)	1.46
MOO$_{\{E\}}$	36.95%(0.42)	**2.57**%(0.02)	0.13(0.09)	**1.50**
MOO$_{\{R\}}$	8.20%(**0.18**)	0.79%(**0.01**)	**0.06**(**0.04**)	0.57

(f) NIFTY 50

	TR	$\mathbb{E}[\mathsf{RoR}]$	Risk	SR
MOO$_{\{T,E\}}$	58.88%(0.64)	3.19%(0.03)	0.10(0.06)	1.21
MOO$_{\{E,R\}}$	28.90%(0.35)	2.31%(0.03)	0.07(0.05)	0.78
MOO$_{\{T,R\}}$	45.46%(0.54)	1.70%(**0.01**)	0.06(0.03)	**1.48**
MOO$_{\{T,E,R\}}$	53.90%(0.62)	3.31%(0.04)	0.09(0.05)	0.77
MOO$_{\{T\}}$	**76.42**%(0.74)	4.14%(0.05)	0.08(0.04)	0.83
MOO$_{\{E\}}$	57.77%(0.51)	**4.59**%(0.04)	0.09(0.04)	1.09
MOO$_{\{R\}}$	18.73%(**0.17**)	1.03%(**0.01**)	**0.04**(**0.01**)	1.33

(g) DJIA

	TR	$\mathbb{E}[\mathsf{RoR}]$	Risk	SR
MOO$_{\{T,E\}}$	74.66%(0.74)	**2.96**%(0.02)	0.08(0.03)	1.83
MOO$_{\{E,R\}}$	38.45%(0.36)	1.67%(0.01)	0.06(0.02)	1.47
MOO$_{\{T,R\}}$	49.49%(0.54)	1.36%(0.01)	0.05(0.02)	**2.12**
MOO$_{\{T,E,R\}}$	65.27%(0.73)	2.34%(0.01)	0.06(0.02)	1.76
MOO$_{\{T\}}$	**81.78**%(0.77)	2.87%(0.02)	0.07(0.03)	1.89
MOO$_{\{E\}}$	57.52%(0.63)	2.90%(0.02)	0.08(0.03)	1.67
MOO$_{\{R\}}$	18.42%(**0.17**)	0.85%(0.01)	**0.03**(0.02)	1.94

(h) NIKKEI 225

	TR	$\mathbb{E}[\mathsf{RoR}]$	Risk	SR
MOO$_{\{T,E\}}$	11.90%(0.17)	1.21%(0.01)	0.14(0.27)	0.99
MOO$_{\{E,R\}}$	12.21%(0.16)	1.58%(0.01)	0.14(0.27)	**1.17**
MOO$_{\{T,R\}}$	26.16%(0.41)	1.59%(0.02)	0.05(0.03)	0.84
MOO$_{\{T,E,R\}}$	29.80%(0.48)	**2.16**%(0.03)	0.06(**0.01**)	0.79
MOO$_{\{T\}}$	**31.97**%(0.45)	2.02%(0.04)	0.04(0.02)	0.54
MOO$_{\{E\}}$	24.41%(0.48)	1.69%(0.03)	0.06(0.04)	0.60
MOO$_{\{R\}}$	8.65%(**0.10**)	0.78%(0.01)	**0.03**(**0.01**)	0.73

(i) S&P 500

	TR	$\mathbb{E}[\mathsf{RoR}]$	Risk	SR
MOO$_{\{T,E\}}$	66.96%(0.51)	**2.78**%(0.01)	0.07(0.03)	2.17
MOO$_{\{E,R\}}$	31.44%(0.24)	1.65%(0.01)	0.05(0.02)	1.97
MOO$_{\{T,R\}}$	52.33%(0.47)	1.39%(0.01)	0.05(0.03)	**2.43**
MOO$_{\{T,E,R\}}$	62.17%(0.51)	2.39%(0.01)	0.07(0.03)	2.12
MOO$_{\{T\}}$	**66.99**%(0.53)	2.65%(0.01)	0.07(0.03)	2.02
MOO$_{\{E\}}$	47.29%(0.36)	2.57%(0.01)	0.07(0.02)	2.27
MOO$_{\{R\}}$	12.74%(**0.13**)	0.58%(0.01)	**0.03**(0.02)	1.35

(j) FTSE 100

	TR	$\mathbb{E}[\mathsf{RoR}]$	Risk	SR
MOO$_{\{T,E\}}$	80.25%(0.38)	4.12%(0.01)	0.07(0.03)	**2.80**
MOO$_{\{E,R\}}$	45.57%(0.28)	2.83%(0.01)	0.05(0.02)	1.93
MOO$_{\{T,R\}}$	68.83%(0.38)	2.46%(0.01)	0.05(0.02)	1.84
MOO$_{\{T,E,R\}}$	80.72%(0.38)	3.64%(0.01)	0.06(0.03)	2.50
MOO$_{\{T\}}$	**90.31**%(0.51)	4.22%(0.02)	0.07(0.04)	1.87
MOO$_{\{E\}}$	74.94%(0.33)	**5.69**%(0.07)	0.08(0.05)	0.87
MOO$_{\{R\}}$	26.13%(**0.18**)	1.32%(0.01)	**0.03**(0.02)	**2.80**

compares two exponential moving averages obtained over a long and a short term period respectively; a zero-crossing strategy based on the comparison between long

Table 10: Statistical test results for average TR, $\mathbb{E}$[RoR], and Risk, according to the non-parametric Friedman test with the Hommel post-hoc test of different MOO algorithms. The order of algorithms in the table corresponds to their ranking according to the Friedman test, with the value in parentheses representing the adjusted p-values. The subscript for each algorithm denotes which metrics were optimised. When more than one metric is present, equal weights have been assigned to each metric. Significant differences at the $\alpha = 5\%$ level between the control algorithm (appearing as the top row in each case) and the remaining algorithms are shown in boldface, indicating that the adjusted p-value is lower than 0.05. Keys: TR: total return, $\mathbb{E}$[RoR]: expected rate of return, SR: Sharpe ratio.

(a) NYSE

TR	$\mathbb{E}$[RoR]	Risk
MOO$_{\{T\}}$(-)	MOO$_{\{T\}}$(-)	MOO$_{\{R\}}$(-)
MOO$_{\{T,E,R\}}$(0.75)	MOO$_{\{T,E,R\}}$(0.76)	MOO$_{\{E,R\}}$(0.13)
MOO$_{\{T,E\}}$(0.72)	MOO$_{\{E\}}$(0.76)	MOO$_{\{T,R\}}$(0.05)
MOO$_{\{T,R\}}$(0.69)	MOO$_{\{T,E\}}$(0.76)	MOO$_{\{T,E,R\}}$(**1.59E-3**)
MOO$_{\{E,R\}}$(0.48)	MOO$_{\{E,R\}}$(0.55)	MOO$_{\{T\}}$(**1.12E-4**)
MOO$_{\{E\}}$(0.47)	MOO$_{\{T,R\}}$(0.55)	MOO$_{\{E\}}$(**4.30E-5**)
MOO$_{\{R\}}$(0.07)	MOO$_{\{R\}}$(**0.03**)	MOO$_{\{T,E\}}$(**1.4E-5**)

(b) NASDAQ

TR	$\mathbb{E}$[RoR]	Risk
MOO$_{\{T\}}$(-)	MOO$_{\{T\}}$(-)	MOO$_{\{R\}}$(-)
MOO$_{\{T,E,R\}}$(0.52)	MOO$_{\{E\}}$(0.81)	MOO$_{\{T,R\}}$(0.44)
MOO$_{\{T,E\}}$(0.24)	MOO$_{\{T,E\}}$(0.74)	MOO$_{\{E,R\}}$(0.34)
MOO$_{\{E\}}$(0.16)	MOO$_{\{T,E,R\}}$(0.48)	MOO$_{\{T,E,R\}}$(0.17)
MOO$_{\{E,R\}}$(0.08)	MOO$_{\{E,R\}}$(0.20)	MOO$_{\{T\}}$(0.08)
MOO$_{\{T,R\}}$(0.06)	MOO$_{\{R\}}$(0.20)	MOO$_{\{T,E\}}$(**0.01**)
MOO$_{\{R\}}$(**0.02**)	MOO$_{\{T,R\}}$(0.20)	MOO$_{\{E\}}$(**0.01**)

(c) TSEC

TR	$\mathbb{E}$[RoR]	Risk
MOO$_{\{T,E\}}$(−)	MOO$_{\{T,E\}}$(−)	MOO$_{\{R\}}$(−)
MOO$_{\{T\}}$(0.66)	MOO$_{\{E\}}$(0.47)	MOO$_{\{E,R\}}$(0.18)
MOO$_{\{T,E,R\}}$(0.30)	MOO$_{\{T,E,R\}}$(0.45)	MOO$_{\{T,R\}}$(0.11)
MOO$_{\{T,R\}}$(0.29)	MOO$_{\{T\}}$(0.45)	MOO$_{\{E\}}$(**0.02**)
MOO$_{\{E\}}$(0.13)	MOO$_{\{E,R\}}$(0.05)	MOO$_{\{T\}}$(**0.01**)
MOO$_{\{E,R\}}$(**0.02**)	MOO$_{\{T,R\}}$(**0.02**)	MOO$_{\{T,E,R\}}$(**3.76E-3**)
MOO$_{\{R\}}$(**4.17E-3**)	MOO$_{\{R\}}$(**2.54E-3**)	MOO$_{\{T,E\}}$(**5.80E-5**)

(d) DAX

TR	$\mathbb{E}$[RoR]	Risk
MOO$_{\{T\}}$(-)	MOO$_{\{T\}}$(-)	MOO$_{\{R\}}$(-)
MOO$_{\{T,E\}}$(0.30)	MOO$_{\{E\}}$(0.44)	MOO$_{\{E,R\}}$(0.23)
MOO$_{\{E\}}$(0.13)	MOO$_{\{T,E\}}$(0.16)	MOO$_{\{T,R\}}$(0.12)
MOO$_{\{T,E,R\}}$(0.06)	MOO$_{\{T,E,R\}}$(**0.02**)	MOO$_{\{T,E,R\}}$(**0.02**)
MOO$_{\{T,R\}}$(**0.04**)	MOO$_{\{E,R\}}$(**0.004**)	MOO$_{\{T\}}$(**0.02**)
MOO$_{\{E,R\}}$(**4E-4**)	MOO$_{\{T,R\}}$(**7.2E-4**)	MOO$_{\{E\}}$(**7.4E-4**)
MOO$_{\{R\}}$(**1.6E-5**)	MOO$_{\{R\}}$(**1.09E-4**)	MOO$_{\{T,E\}}$(**2.01E-4**)

(e) RUSSELL 2000

TR	$\mathbb{E}$[RoR]	Risk
MOO$_{\{T\}}$(-)	MOO$_{\{T,E\}}$(-)	MOO$_{\{R\}}$(-)
MOO$_{\{T,E,R\}}$(0.46)	MOO$_{\{T,E,R\}}$(1.00)	MOO$_{\{E,R\}}$(0.26)
MOO$_{\{E\}}$(0.43)	MOO$_{\{T\}}$(1.00)	MOO$_{\{T,R\}}$(0.06)
MOO$_{\{T,E\}}$(0.31)	MOO$_{\{E\}}$(1.00)	MOO$_{\{T,E,R\}}$(**0.003**)
MOO$_{\{E,R\}}$(0.31)	MOO$_{\{T,R\}}$(0.23)	MOO$_{\{T\}}$(**0.002**)
MOO$_{\{T,R\}}$(0.31)	MOO$_{\{E,R\}}$(0.23)	MOO$_{\{T,E\}}$(**1.6E-4**)
MOO$_{\{R\}}$(0.06)	MOO$_{\{R\}}$(0.10)	MOO$_{\{E\}}$(**8.5E-5**)

(f) NIFTY 50

TR	$\mathbb{E}$[RoR]	Risk
MOO$_{\{T\}}$(-)	MOO$_{\{E\}}$(-)	MOO$_{\{R\}}$(-)
MOO$_{\{E\}}$(0.22)	MOO$_{\{T,E\}}$(0.34)	MOO$_{\{T,R\}}$(0.13)
MOO$_{\{T,E\}}$(0.12)	MOO$_{\{T\}}$(0.29)	MOO$_{\{E,R\}}$(**0.04**)
MOO$_{\{T,E,R\}}$(0.06)	MOO$_{\{T,E,R\}}$(0.14)	MOO$_{\{T\}}$(**0.003**)
MOO$_{\{T,R\}}$(**0.02**)	MOO$_{\{T,R\}}$(**0.012**)	MOO$_{\{T,E,R\}}$(**0.002**)
MOO$_{\{R\}}$(**8E-4**)	MOO$_{\{E,R\}}$(**0.01**)	MOO$_{\{T,E\}}$(**5E-5**)
MOO$_{\{E,R\}}$(**8E-4**)	MOO$_{\{R\}}$(**0.002**)	MOO$_{\{E\}}$(**5E-5**)

(g) DJIA

TR	$\mathbb{E}$[RoR]	Risk
MOO$_{\{T\}}$(-)	MOO$_{\{E\}}$(-)	MOO$_{\{R\}}$(-)
MOO$_{\{T,E\}}$(0.26)	MOO$_{\{T,E\}}$(0.44)	MOO$_{\{T,R\}}$(0.08)
MOO$_{\{T,E,R\}}$(0.07)	MOO$_{\{T\}}$(0.41)	MOO$_{\{E,R\}}$(**0.009**)
MOO$_{\{E\}}$(**0.003**)	MOO$_{\{T,E,R\}}$(0.13)	MOO$_{\{T,E,R\}}$(**0.006**)
MOO$_{\{T,R\}}$(**0.001**)	MOO$_{\{E,R\}}$(**0.03**)	MOO$_{\{T\}}$(**0.001**)
MOO$_{\{E,R\}}$(**6E-4**)	MOO$_{\{T,R\}}$(**0.004**)	MOO$_{\{E\}}$(**1.1E-4**)
MOO$_{\{R\}}$(**1.6E-5**)	MOO$_{\{R\}}$(**5.9E-4**)	MOO$_{\{T,E\}}$(**1.1E-4**)

(h) NIKKEI 225

TR	$\mathbb{E}$[RoR]	Risk
MOO$_{\{T\}}$(-)	MOO$_{\{T,E,R\}}$(-)	MOO$_{\{R\}}$(-)
MOO$_{\{T,E,R\}}$(0.69)	MOO$_{\{T\}}$(0.92)	MOO$_{\{T\}}$(**0.02**)
MOO$_{\{T,R\}}$(0.62)	MOO$_{\{E,R\}}$(0.81)	MOO$_{\{T,R\}}$(**0.01**)
MOO$_{\{E,R\}}$(0.26)	MOO$_{\{T,E\}}$(0.663603)	MOO$_{\{T,E,R\}}$(**0.001**)
MOO$_{\{T,E\}}$(0.26)	MOO$_{\{E\}}$(0.53)	MOO$_{\{E,R\}}$(**0.001**)
MOO$_{\{E\}}$(0.20)	MOO$_{\{T,R\}}$(0.45)	MOO$_{\{T,E\}}$(**6E-4**)
MOO$_{\{R\}}$(0.20)	MOO$_{\{R\}}$(0.09)	MOO$_{\{E\}}$(**5.3E-5**)

(i) S&P 500

TR	$\mathbb{E}$[RoR]	Risk
MOO$_{\{T\}}$(-)	MOO$_{\{T,E\}}$(-)	MOO$_{\{R\}}$(-)
MOO$_{\{T,E\}}$(0.28)	MOO$_{\{E\}}$(0.44)	MOO$_{\{T,R\}}$(0.08)
MOO$_{\{T,E,R\}}$(0.18)	MOO$_{\{T\}}$(0.33)	MOO$_{\{E,R\}}$(0.06)
MOO$_{\{E\}}$(**0.04**)	MOO$_{\{T,E,R\}}$(0.17)	MOO$_{\{T,E,R\}}$(**0.001**)
MOO$_{\{T,R\}}$(**0.009**)	MOO$_{\{E,R\}}$(**0.02**)	MOO$_{\{T\}}$(**2.1E-4**)
MOO$_{\{E,R\}}$(**0.005**)	MOO$_{\{T,R\}}$(**0.005**)	MOO$_{\{E\}}$(**1.2E-4**)
MOO$_{\{R\}}$(**5.8E-5**)	MOO$_{\{R\}}$(**9E-5**)	MOO$_{\{T,E\}}$(**1.2E-4**)

(j) FTSE 100

TR	$\mathbb{E}$[RoR]	Risk
MOO$_{\{T\}}$(-)	MOO$_{\{T,E\}}$(-)	MOO$_{\{R\}}$(-)
MOO$_{\{T,E,R\}}$(0.20)	MOO$_{\{E\}}$(0.26	MOO$_{\{T,R\}}$(0.09)
MOO$_{\{T,E\}}$(0.20)	MOO$_{\{T\}}$(0.17)	MOO$_{\{E,R\}}$(**0.03**)
MOO$_{\{E\}}$(0.14)	MOO$_{\{T,E,R\}}$(0.12)	MOO$_{\{T,E,R\}}$(**0.001**)
MOO$_{\{T,R\}}$(**0.02**)	MOO$_{\{E,R\}}$(**0.02**)	MOO$_{\{T\}}$(**4E-4**)
MOO$_{\{E,R\}}$(**4E-4**)	MOO$_{\{T,R\}}$(**0.001**)	MOO$_{\{E\}}$(**2.3E-5**)
MOO$_{\{R\}}$(**4.2E-5**)	MOO$_{\{R\}}$(**4.6E-5**)	MOO$_{\{T,E\}}$(**1.6E-5**)

and short-term values for the On Balance Volume (OBV) indicator; and a threshold-based strategy based on the Momentum (MTM) indicator, which generates a buy signal when the MTM exceeds a specified positive threshold, and a sell signal when it falls below a specified negative threshold respectively (which, in our case, was made to coincide with the chosen threshold for the DC framework). All TA benchmarks were fine-tuned to choose optimal values for long/short period hyperparameters in the set $\{5, 10, 30, 50\}$; fine-tuning was performed in a similar manner to that described for MOO and SOO in section 5.4.

From Table 8, we concluded that $\mathsf{MOO}_{\{\mathsf{T}\}}$, $\mathsf{MOO}_{\{\mathsf{E}\}}$, and $\mathsf{MOO}_{\{\mathsf{R}\}}$ rank first at their corresponding Friedman tests. We therefore bring forward these algorithms to be compared against the TA indicators. In addition to MOO3 algorithms optimising a single metric, we are also interested in bringing forward for comparison, algorithms that optimised more than one metric; we thus looked at the second rank of each of Tables 8a, 8b, and 8c, and we will also compare $\mathsf{MOO}_{\{\mathsf{T,E}\}}$ and $\mathsf{MOO}_{\{\mathsf{T,R}\}}$ against the TA benchmarks.

Table 11 presents the results across total return, expected rate of return, risk, and portfolio Sharpe ratio. As we can observe, in all cases the MOO3 algorithms have significantly improved values over MACD, OBV, and MTM. As previously seen, $\mathsf{MOO}_{\{\mathsf{T}\}}$ and $\mathsf{MOO}_{\{\mathsf{E}\}}$ offer the best performance for total return and expected rate of return respectively, and generally have a good overall performance on maximising returns; however, this comes at the cost of higher risk. On the other hand, $\mathsf{MOO}_{\{\mathsf{R}\}}$ minimises risk effectively, but at the cost of lower returns. Lastly, $\mathsf{MOO}_{\{\mathsf{T,R}\}}$ offers a balanced approach with a high Sharpe ratio, indicating good risk-adjusted performance.

Figure 7 further illustrates the findings of Table 11 in the form of a box-plot. In the case of TR and $\mathbb{E}[\mathsf{RoR}]$, all MOO algorithms have boxes that are above the zero line (i.e. not only the median, but also the first quartile of the distribution of TRs is positive in the case of MOO algorithms). Whereas, the three standard TA-indicator based

Table 11: Summary statistics of the three-objective optimisation algorithms and technical analysis indicators. We use boldface for the best values for each measure.

Measure	Algorithm							
	Total return							
	$\mathsf{MOO}_{\{\mathsf{T}\}}$	$\mathsf{MOO}_{\{\mathsf{E}\}}$	$\mathsf{MOO}_{\{\mathsf{R}\}}$	$\mathsf{MOO}_{\{\mathsf{T,E}\}}$	$\mathsf{MOO}_{\{\mathsf{T,R}\}}$	MACD	OBV	MTM
Average	**61.33%**	48.29%	15.15%	51.90%	41.58%	13.99%	2.63%	16.25%
Median	**41.68%**	32.81%	10.06%	37.88%	24.18%	2.22%	1.73%	2.03%
Standard deviation	0.67	0.69	**0.24**	0.61	0.52	0.58	0.43	0.81
Max	341.71%	544.98%	152.06%	324.26%	258.04%	360.11%	148.28%	**659.06%**
Min	-44.63%	-44.89%	**-32.11%**	-42.72%	-37.47%	-90.38%	-177.72%	-50.62%
	Rate of return							
	$\mathsf{MOO}_{\{\mathsf{T}\}}$	$\mathsf{MOO}_{\{\mathsf{E}\}}$	$\mathsf{MOO}_{\{\mathsf{R}\}}$	$\mathsf{MOO}_{\{\mathsf{T,E}\}}$	$\mathsf{MOO}_{\{\mathsf{T,R}\}}$	MACD	OBV	MTM
Average	3.28%	**3.38%**	0.99%	2.76%	1.63%	1.71%	0.77%	0.95%
Median	2.52%	**2.62%**	0.86%	2.38%	1.44%	0.14%	0.18%	0.22%
Standard deviation	0.05	0.04	0.02	0.02	**0.01**	0.06	0.05	0.06
Max	**44.36%**	25.92%	12.24%	13.07%	6.80%	36.01%	35.65%	36.28%
Min	-4.03%	-4.28%	-1.98%	-2.98%	**-1.23%**	-9.04%	-11.11%	-22.32%
	Risk							
	$\mathsf{MOO}_{\{\mathsf{T}\}}$	$\mathsf{MOO}_{\{\mathsf{E}\}}$	$\mathsf{MOO}_{\{\mathsf{R}\}}$	$\mathsf{MOO}_{\{\mathsf{T,E}\}}$	$\mathsf{MOO}_{\{\mathsf{T,R}\}}$	MACD	OBV	MTM
Average	0.08	0.09	**0.05**	0.09	0.06	0.12	0.12	0.09
Median	0.06	0.07	**0.03**	0.07	0.05	0.07	0.09	0.06
Standard deviation	0.06	0.07	0.05	0.06	**0.04**	0.22	0.12	0.12
Max	0.43	0.49	0.42	0.41	**0.29**	1.58	0.76	1.15
Min	0.01	0.01	0.01	0.01	0.01	**0.00**	**0.00**	**0.00**
	Sharpe ratio							
	$\mathsf{MOO}_{\{\mathsf{T}\}}$	$\mathsf{MOO}_{\{\mathsf{E}\}}$	$\mathsf{MOO}_{\{\mathsf{R}\}}$	$\mathsf{MOO}_{\{\mathsf{T,E}\}}$	$\mathsf{MOO}_{\{\mathsf{T,R}\}}$	MACD	OBV	MTM
SR	0.66	0.76	0.65	1.10	**1.12**	0.24	0.06	0.17

strategy benchmarks demonstrate medians very close to zero, and in fact negative first-quartiles. When looking at Risk, we can observe that $\mathsf{MOO}_{\{\mathsf{T}\}}$, $\mathsf{MOO}_{\{\mathsf{E}\}}$, and $\mathsf{MOO}_{\{\mathsf{T,E}\}}$ obtain similar values to the benchmarks, but all are outperformed by $\mathsf{MOO}_{\{\mathsf{R}\}}$ and $\mathsf{MOO}_{\{\mathsf{T,R}\}}$, both in terms of median risk values, as well as in terms of demonstrating a more concentrated risk distribution, and fewer outliers.

The above findings are further supported by the Friedman tests (one per metric) presented in Table 12. In all three tables, we can observe that the first rank MOO3 algorithm statistically and significantly outperforms all other algorithms at the 5% significance level. In addition, the majority of the multi-objective optimisation algorithms are able to outrank the trading strategies derived by using TA indicators.

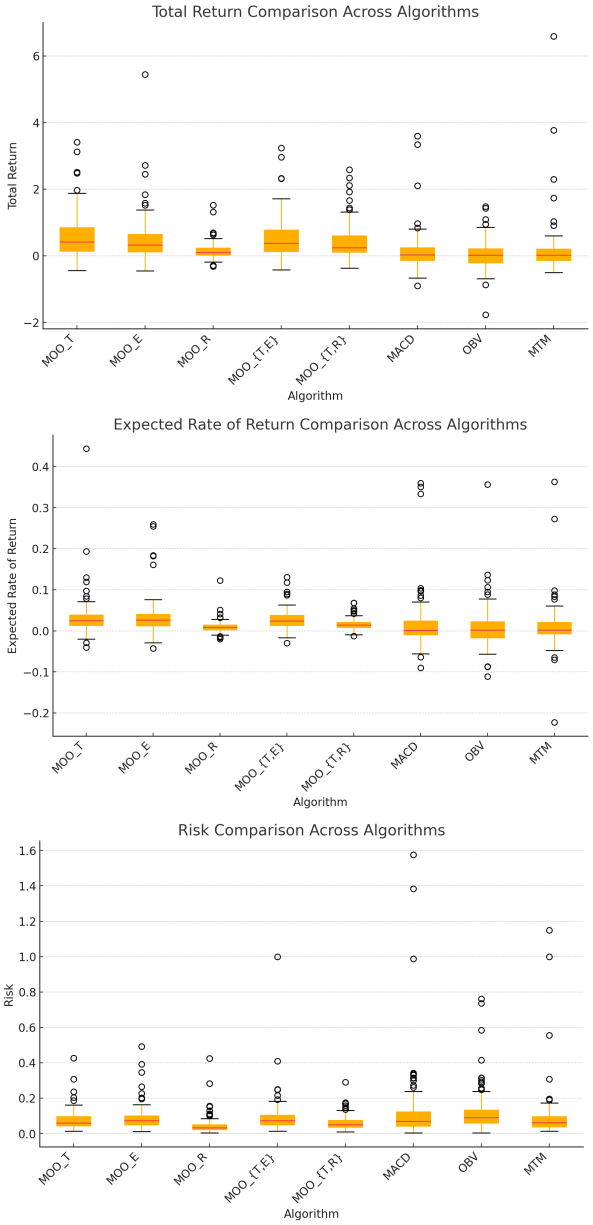


Fig 7 Box plot of MOO algorithms and three standard TA-indicator based strategies in terms of TR, $\mathbb{E}[\mathsf{RoR}]$, and Risk

Table 12: Statistical test results between MOO3 algorithms and technical analysis indicators for average TR, $\mathbb{E}$[RoR], and Risk, according to the non-parametric Friedman test with the Hommel post-hoc test. The subscript for each algorithm denotes which metrics were optimised. When more than one metric is present, equal weights have been assigned to each metric. Significant differences at the $\alpha = 5\%$ level between the control algorithm (appearing as the top row in each case) and the remaining algorithms are shown in boldface, indicating that the adjusted p-value is lower than 0.05.

(a) Total Return

	Avg Rank	Adj p_{Homm}
$\mathsf{MOO}_{\{\mathsf{T}\}}$	2.17	-
$\mathsf{MOO}_{\{\mathsf{T,E}\}}$	**3.07**	2.48E-03
$\mathsf{MOO}_{\{\mathsf{E}\}}$	**3.46**	4.11E-05
$\mathsf{MOO}_{\{\mathsf{T,R}\}}$	**3.78**	5.93E-07
$\mathsf{MOO}_{\{\mathsf{R}\}}$	**5.48**	6.32E-22
MACD	**5.76**	3.36E-25
MTM	**6.08**	4.25E-29
OBV	**6.18**	3.98E-30

(b) Expected rate of Return

	Avg Rank	Adj p_{Homm}
$\mathsf{MOO}_{\{\mathsf{E}\}}$	2.91	-
$\mathsf{MOO}_{\{\mathsf{T,E}\}}$	2.94	3.00E-01
$\mathsf{MOO}_{\{\mathsf{T}\}}$	3.13	1.99E-01
$\mathsf{MOO}_{\{\mathsf{T,R}\}}$	**4.53**	8.71E-07
MACD	**5.38**	2.48E-13
$\mathsf{MOO}_{\{\mathsf{R}\}}$	**5.61**	2.65E-15
MTM	**5.67**	1.24E-15
OBV	**5.79**	2.18E-16

(c) Risk

	Avg Rank	Adj p_{Homm}
$\mathsf{MOO}_{\{\mathsf{R}\}}$	1.60	-
$\mathsf{MOO}_{\{\mathsf{T,R}\}}$	**3.33**	1.72E-07
$\mathsf{MOO}_{\{\mathsf{T}\}}$	**4.49**	2.69E-17
MTM	**4.50**	2.08E-17
MACD	**5.11**	2.92E-24
$\mathsf{MOO}_{\{\mathsf{E}\}}$	**5.47**	1.27E-28
$\mathsf{MOO}_{\{\mathsf{T,E}\}}$	**5.59**	5.27E-30
OBV	**5.87**	1.50E-33

6.6 Comparison of MOO algorithms to a Transformer-based architecture

In the previous section, we observed that our proposed multi-objective GP algorithm outperformed three standard TA-indicator-based strategies. In this section, we extend the comparison to the state-of-the-art Transformer-based model described in Section 5.2. As shown in Table 13, the proposed MOO algorithms consistently achieve higher TR values than the Transformer, which, despite achieving the highest maximum TR, shows poor consistency with a significantly lower median and higher variability. In contrast, the Transformer achieves the lowest risk on average, comparable to $\mathsf{MOO}_{\{\mathsf{R}\}}$,

Table 13: Summary statistics for the three-objective optimisation algorithms and the Transformer benchmark. Best values per metric appear in boldface.

	Total return					
	MOO$_{\{T\}}$	MOO$_{\{E\}}$	MOO$_{\{R\}}$	MOO$_{\{T,E\}}$	MOO$_{\{T,R\}}$	Transformer
Average	**61.33%**	48.29%	15.15%	51.90%	41.58%	8.81%
Median	**41.68%**	32.81%	10.06%	37.88%	24.18%	0.96%
Standard deviation	0.67	0.69	**0.24**	0.61	0.52	0.77
Max	341.71%	544.98%	152.06%	324.26%	258.04%	**762.04%**
Min	-44.63%	-44.89%	-32.11%	-42.72%	-37.47%	**-78.30%**
	Rate of return					
	MOO$_{\{T\}}$	MOO$_{\{E\}}$	MOO$_{\{R\}}$	MOO$_{\{T,E\}}$	MOO$_{\{T,R\}}$	Transformer
Average	3.28%	**3.38%**	0.99%	2.76%	1.63%	0.68%
Median	2.52%	**2.62%**	0.86%	2.38%	1.44%	0.51%
Standard deviation	0.05	0.04	0.02	0.02	**0.01**	**0.01**
Max	**44.36%**	25.92%	12.24%	13.07%	6.80%	6.91%
Min	-4.03%	-4.28%	-1.98%	-2.98%	**-1.23%**	-2.45%
	Risk					
	MOO$_{\{T\}}$	MOO$_{\{E\}}$	MOO$_{\{R\}}$	MOO$_{\{T,E\}}$	MOO$_{\{T,R\}}$	Transformer
Average	0.08	0.09	**0.05**	0.09	0.06	**0.05**
Median	0.06	0.07	**0.03**	0.07	0.05	0.04
Standard deviation	0.06	0.07	0.05	0.06	**0.04**	**0.04**
Max	0.43	0.49	0.42	0.41	0.29	**0.22**
Min	0.01	0.01	0.01	0.01	0.01	0.01
	Sharpe ratio					
	MOO$_{\{T\}}$	MOO$_{\{E\}}$	MOO$_{\{R\}}$	MOO$_{\{T,E\}}$	MOO$_{\{T,R\}}$	Transformer
SR	0.66	0.76	0.65	1.10	**1.12**	0.50

though it is still statistically outperformed by the latter. However, this comes at a trade-off: the Transformer has the worst Sharpe Ratio (0.50), whereas MOO$_{\{T,R\}}$ and MOO$_{\{T,E\}}$ demonstrate significantly better return-risk trade-offs. These findings are further confirmed by the Friedman test results in Table 14, where the Transformer ranks last for TR and $\mathbb{E}$[RoR] and second for risk, though still significantly weaker than MOO$_{\{R\}}$ (p = 1.15E-05).

6.7 Buy and hold

We now compare the proposed MOO3 algorithms with the buy-and-hold strategy. A critical aspect to note is that the buy-and-hold strategy inherently involves a single transaction over the entire period under study, where we buy one unit of stock on

Table 14: Statistical test results between MOO3 algorithms and Transformer for average TR, $\mathbb{E}$[RoR], and Risk, according to the non-parametric Friedman test with the Hommel post-hoc test. The subscript for each algorithm denotes which metrics were optimised. When more than one metric is present, equal weights have been assigned to each metric. Significant differences at the $\alpha = 5\%$ level between the control algorithm (appearing as the top row in each case) and the remaining algorithms are shown in boldface, indicating that the adjusted p-value is lower than 0.05.

(a) Total Return

	Avg Rank	Adj p_{Homm}
$\mathsf{MOO}_{\{\mathsf{T}\}}$	1.83	-
$\mathsf{MOO}_{\{\mathsf{T,E}\}}$	**2.65**	2.33E-04
$\mathsf{MOO}_{\{\mathsf{E}\}}$	**3.10**	1.56E-07
$\mathsf{MOO}_{\{\mathsf{T,R}\}}$	**3.34**	1.33E-09
$\mathsf{MOO}_{\{\mathsf{R}\}}$	**4.71**	3.86E-27
Transformer	**5.35**	1.40E-37

(b) Expected rate of Return

	Avg Rank	Adj p_{Homm}
$\mathsf{MOO}_{\{\mathsf{E}\}}$	2.41	-
$\mathsf{MOO}_{\{\mathsf{T,E}\}}$	2.44	2.36E-01
$\mathsf{MOO}_{\{\mathsf{T}\}}$	2.52	1.97E-01
$\mathsf{MOO}_{\{\mathsf{T,R}\}}$	**3.78**	4.87E-08
$\mathsf{MOO}_{\{\mathsf{R}\}}$	**4.80**	7.01E-20
Transformer	**5.02**	1.85E-22

(c) Risk

	Avg Rank	Adj p_{Homm}
$\mathsf{MOO}_{\{\mathsf{R}\}}$	1.55	-
Transformer	**2.53**	1.15E-05
$\mathsf{MOO}_{\{\mathsf{T,R}\}}$	**3.11**	1.53E-10
$\mathsf{MOO}_{\{\mathsf{T}\}}$	**4.01**	2.89E-21
$\mathsf{MOO}_{\{\mathsf{E}\}}$	**4.80**	1.93E-33
$\mathsf{MOO}_{\{\mathsf{T,E}\}}$	**4.97**	4.04E-36

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 MOO GPs make several), it is fairer to compare them across the total return over the test set period rather than in terms of expected rate of return.

Table 15 presents the performance metrics of the MOO3 algorithms alongside the buy-and-hold strategy. We can observe that buy-and-hold has a strong average performance (mainly due to outliers), while its median value of 11.44% is only able to outperform $\mathsf{MOO}_{\{\mathsf{R}\}}$, which is understandable, given that the latter does not optimise total return. All other algorithms have significantly higher average and median values. These results are also confirmed by the Friedman test presented in Table 16, which show that buy-and-hold is statistically and significantly outperformed by $\mathsf{MOO}_{\{\mathsf{T}\}}$.

Table 15: Summary statistics for the three-objective optimisation algorithms and buy-and-hold in terms of total return. The best values per metric appear in boldface.

Algorithm	$\mathsf{MOO}_{\{\mathsf{T}\}}$	$\mathsf{MOO}_{\{\mathsf{E}\}}$	$\mathsf{MOO}_{\{\mathsf{R}\}}$	$\mathsf{MOO}_{\{\mathsf{T,E}\}}$	$\mathsf{MOO}_{\{\mathsf{T,R}\}}$	Buy-and-hold
Average	**61.33%**	48.29%	15.15%	51.90%	41.58%	41.11%
Median	**41.68%**	32.81%	10.06%	37.88%	24.18%	11.44%
Standard deviation	0.67	0.69	**0.24**	0.61	0.52	1.81
Max	341.71%	544.98%	152.06%	324.26%	258.04%	**1753.05%**
Min	-44.63%	-44.89%	-32.11%	-42.72%	-37.47%	**-89.62%**

Table 16: Statistical test results between MOO3 algorithms and buy-and-hold strategy in terms of average Total Return, according to the non-parametric Friedman test with the Hommel post-hoc test. The subscript for each algorithm denotes which metrics were optimised. When more than one metric is present, equal weights have been assigned to each metric. Significant differences at the $\alpha = 5\%$ level between the control algorithm (appearing as the top row) and the other algorithms are shown in boldface, indicating that the adjusted p-value is lower than 0.05.

Algorithm	Avg rank	Adj p_{Homm}
$\mathsf{MOO}_{\{\mathsf{T}\}}$ (c)	1.97	-
$\mathsf{MOO}_{\{\mathsf{T,E}\}}$	**2.83**	1.24E-04
$\mathsf{MOO}_{\{\mathsf{E}\}}$	**3.26**	1.14E-07
$\mathsf{MOO}_{\{\mathsf{T,R}\}}$	**3.51**	6.24E-10
Buy-and-hold	**4.40**	2.88E-20
$\mathsf{MOO}_{\{\mathsf{R}\}}$	**5.00**	2.44E-20

6.8 Algorithmic complexity

To discuss the potential scalability of the approach, this section demonstrates the computational cost of the proposed GP-based algorithm throughout its process and evolutionary operations. To calculate the algorithmic complexity of the GP-based algorithm, some related parameters need to be introduced as the number of optimisation objectives (M), the population size (P), maximum depth of tree (N), dataset period days (m), and tournament size (T). The main process of the proposed GP-based algorithm could be broken down into the following steps:

1. Population initialisation:

   The initialisation of one individual has a computational complexity $\mathcal{O}(N)$ with an N maximum depth of the tree. As the population size is P, the initialisation is repeated P times, tuning out a complexity of $\mathcal{O}(PN)$.

2. Fitness function calculation:

   Similarly, calculating fitness functions for one individual has a complexity $\mathcal{O}(m)$ throughout the whole dataset period of m days. With the P population size, the computational complexity of fitness function calculation is $\mathcal{O}(Pm)$.

3. Non-dominated Sorting:

   In this process, each individual needs to be compared with the rest of the population to determine dominance. In the worst case, where no individual is dominated, the complexity is $\mathcal{O}(P^2)$. Taking M objectives into account, the non-dominated sorting requires $\mathcal{O}(MP^2)$ complexity.

4. Crowding distance calculation:

   For each front, individuals are sorted in terms of each objective. Per the objective, the sorting takes $\mathcal{O}(P \log P)$, leading to an overall complexity of $\mathcal{O}(MP \log P)$.

5. Tournament selection:

   With a tournament size of T and a population size of P, it requires $\mathcal{O}(PT)$ complexity.

6. Genetic operators:

   In the proposed algorithms, subtree crossover and point mutation are applied. For both operators, a node needs to be selected randomly. It requires $\mathcal{O}(N)$ as the algorithm to traverse the whole tree. Furthermore, the subtree needs to be traversed during subtree crossover, which takes another $\mathcal{O}(N)$ in the worst condition. The process takes P times. So the complexity of subtree crossover and point mutation is $\mathcal{O}(2PN) = \mathcal{O}(PN)$ and $\mathcal{O}(PN)$.

7. Population replacement:

   After applying crossover and mutation, the proposed algorithms created a new population containing P offspring. The next step is to select the top P individuals from the combination of the new population and the old population, resulting in $2P$ individuals. It also requires non-dominated sorting ($\mathcal{O}(M(2P)^2) = \mathcal{O}(MP^2)$) and crowding distance calculation ($\mathcal{O}(2P\log(2P)) = \mathcal{O}(P\log P)$) of the combining population. So, the complexity of population replacement is $\mathcal{O}(MP^2 + P\log P)$

In conclusion, the overall complexity of the proposed GP-based algorithm is $\mathcal{O}(PN + Pm + MP^2 + MP\log P + PT + PN + PN + MP^2 + P\log P)$, which is equivalent to $\mathcal{O}(PN + Pm + MP^2 + MP\log P + PT)$.

6.9 Real-world scalability

Although training the proposed GP-based algorithm is computationally expensive, applying predictions is not. This is because once the algorithm has been trained and a trading model has been obtained, executing it can happen in a fraction of a second.

Nevertheless, given the market's inherent volatility and instability, it may be necessary to periodically re-train the algorithm to capture new changes in market conditions. To avoid long periods of retraining the GP algorithm, parallelisation should be used. Evolutionary algorithms are well-suited for parallelisation since each individual in the population is produced and evaluated independently. Previous studies have demonstrated that parallel implementation can achieve speedups of up to 21 times (Brookhouse et al, 2014).

7 Conclusions

In this study, we explored the integration of directional changes, genetic programming, and multi-objective optimisation (MOO) to develop and evaluate advanced algorithmic trading strategies. The proposed MOO3 algorithm, using the NSGA-II algorithm, optimised three objectives: total return, expected rate of return, and risk. Furthermore, we devised a novel aggregate metric, dubbed here the 'modified Sharpe Ratio', allowing us to designate a final solution from the Pareto front, by means of adjustable weights reflecting trader preference among the different objectives. Our experimental results, conducted on 110 datasets from 10 different international markets, demonstrate the superiority of the MOO3 algorithm over single-objective optimisation (SOO) methods employing the same aggregate metric.

The MOO3 algorithm effectively generated a diverse set of Pareto-optimal solutions that provided optimal trade-offs among the three objectives. This was evidenced by higher mean and median total returns, expected rate of returns, and Sharpe ratios across various weight setups when compared to SOO. Statistical analysis further reinforced the findings. Kolmogorov-Smirnov tests indicated significant differences between MOO3 and SOO in several cases, while the non-parametric Friedman test with Hommel's post-hoc analysis showed that MOO3 algorithms focusing on individual objectives performed best, followed by those combining multiple objectives. These results highlight the benefits of considering multiple objectives under a MOO framework, particularly in balancing return and risk. Moreover, MOO3 algorithms were benchmarked against traditional technical analysis indicators (MACD, OBV, and MTM),the state-of-the-art Transformer model, and the buy-and-hold strategy. The MOO3 algorithms consistently outperformed these benchmarks, demonstrating their robustness and efficiency in optimising trading strategies.

In summary, this study provides compelling evidence that multi-objective optimisation under the directional changes framework offers significant improvements over single-objective optimisation. The MOO3 algorithm not only produces superior trading strategies but is also able to consider diverse investor preferences through the use of the modified Sharpe Ratio aggregate metric, making it a valuable tool for algorithmic trading.

Future work in this area could explore several promising directions to further enhance the effectiveness and robustness of algorithmic trading under the DC framework. One potential direction is the incorporation of additional objectives, such as the number of trades that each trading strategy can perform. Another research direction could be the integration of alternative data sources, such as social media sentiment, which would enrich the feature set used in the genetic programming algorithm. Finally, the weighting approach used in the context of the modified Sharpe Ratio aggregate metric, while allowing traders to state their relative preferences for each objective, is still somewhat mathematical in nature; developing / assessing more intuitive methods for designating a single, final trading strategy from the Pareto front could allow traders increased flexibility and clarity when stating their relative preferences over the multiple objectives.

Declarations

Author contributions

Xinpeng Long was responsible for conceptualisation, methodology, software, validation, formal analysis, investigation, resources, data curation, writing the original draft, and visualisation. Michael Kampouridis was responsible for conceptualisation, methodology, software, formal Analysis, reviewing and editing the manuscript, and supervision. Tasos Papastylianou was responsible for methodology, reviewing and editing the manuscript, and visualisation.

Funding

The authors did not receive support from any organization for the submitted work.

Competing interests

The authors have no competing interests to declare that are relevant to the content of this article.

References

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

Adegboye A, Kampouridis M, Otero F (2021) Improving trend reversal estimation in forex markets under a directional changes paradigm with classification algorithms. International Journal of Intelligent Systems 36(12):7609—7640

Adegboye A, Kampouridis M, Otero F (2023) Algorithmic trading with directional changes. Artificial Intelligence Review 56(6):5619—5644

de Almeida R, Reynoso-Meza G, Steiner MTA (2016) Multi-objective optimization approach to stock market technical indicators. In: 2016 IEEE congress on evolutionary computation (CEC), IEEE, pp 3670—3677

Aloud M (2012) 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

Aloud ME (2015) Directional-change event trading strategy: Profit-maximizing learning strategy. The Seventh International Conference on Advanced Cognitive Technologies and Applications 2015 p 134

Aloud ME (2016a) Profitability of directional change based trading strategies: The case of saudi stock market. International Journal of Economics and Financial Issues 6(1):87—95

Aloud ME (2016b) Time series analysis indicators under directional changes: The case of saudi stock market. International Journal of Economics and Financial Issues 6(1):55—64

Aloud ME, Fasli M (2016) Exploring trading strategies and their effects in the foreign exchange market. Computational Intelligence 33(2)

Ao H, Tsang E (2019) Trading algorithms built with directional changes. In: 2019 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr), IEEE, pp 1—7

Atiah FD, Helbig M (2019) Effects of decision models on dynamic multi-objective optimization algorithms for financial markets. In: 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE, pp 762—770

Bakhach A, Tsang E, Ng WL, et al (2016a) Backlash agent: A trading strategy based on directional change. In: 2016 IEEE symposium series on computational intelligence (SSCI), IEEE, pp 1—9

Bakhach A, Tsang EP, Jalalian H (2016b) Forecasting directional changes in the fx markets. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, pp 1—8

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

Brookhouse J, Otero FE, Kampouridis M (2014) Working with opencl to speed up a genetic programming financial forecasting algorithm: initial results. In: Proceedings of the companion publication of the 2014 annual conference on genetic and evolutionary computation, pp 1117–1124

Chen CC, Chang CC, Sun EW, et al (2022) Optimal decision of dynamic wealth allocation with life insurance for mitigating health risk under market incompleteness. European Journal of Operational Research 300(2):727–742

Christodoulaki E, Kampouridis M, Kanellopoulos P (2022) Technical and sentiment analysis in financial forecasting with genetic programming. In: 2022 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr), IEEE, pp 1–8

Christodoulaki E, Kampouridis M, Kyropoulou M (2023) Enhanced strongly typed genetic programming for algorithmic trading. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp 1055–1063

Deb K, Pratap A, Agarwal S, et al (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation 6(2):182—197

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. The Journal of Machine learning research 7:1–30

Deprez N, Frömmel M (2024) Are simple technical trading rules profitable in bitcoin markets? International Review of Economics & Finance 93:858–874

Fong S, Si YW, Tai J (2012) Trend following algorithms in automated derivatives market trading. Expert systems with applications 39(13):11378–11390

Gal Y, Ghahramani Z (2016) Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning, PMLR, pp 1050–1059

Gao Y, Wang R, Zhou E (2021) Stock prediction based on optimized lstm and gru models. Scientific Programming 2021(1):4055281

Garcia S, Herrera F (2008) An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. Journal of machine learning research 9(12)

Glattfelder JB, Dupuis A, Olsen RB (2011) Patterns in high-frequency fx data: discovery of 12 empirical scaling laws. Quantitative Finance 11(4):599—614

Guillaume DM, Dacorogna MM, Davé RR, et al (1997) 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

Gypteau J, Otero FE, Kampouridis M (2015) Generating directional change based trading strategies with genetic programming. In: Applications of Evolutionary Computation: 18th European Conference, EvoApplications 2015, Copenhagen, Denmark, April 8-10, 2015, Proceedings 18, Springer, pp 267—278

Karasu S, Altan A, Bekiros S, et al (2020) A new forecasting model with wrapper-based feature selection approach using multi-objective optimization technique for chaotic crude oil time series. Energy 212:118750

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

Kim Y, Enke D (2016) Developing a rule change trading system for the futures market using rough set analysis. Expert Systems with Applications 59:165—173

Lai WN, Chen YT, Sun EW (2021) Comonotonicity and low volatility effect. Annals of Operations Research 299(1):1057–1099

Leung MF, Wang J (2020) Minimax and biobjective portfolio selection based on collaborative neurodynamic optimization. IEEE Transactions on Neural Networks and Learning Systems 32(7):2825–2836

Long X, Kampouridis M (2024) α-dominance two-objective optimization genetic programming for algorithmic trading under a directional changes environment. In: 2024 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr), IEEE, pp 1–8

Long X, Kampouridis M, Jarchi D (2022a) 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, pp 01—08

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

Long X, Kampouridis M, Kanellopoulos P (2023) Multi-objective optimisation and genetic programming for trading by combining directional changes and technical indicators. In: 2023 IEEE Congress on Evolutionary Computation (CEC), IEEE, pp 1–8

Rayment G, Kampouridis M (2023) High frequency trading with deep reinforcement learning agents under a directional changes sampling framework. In: 2023 IEEE

Symposium Series on Computational Intelligence (SSCI), IEEE, pp 387—394

Rayment G, Kampouridis M (2024) Enhancing high-frequency trading with deep reinforcement learning using advanced positional awareness under a directional changes paradigm. In: 2024 International Conference on Machine Learning and Applications (ICMLA), IEEE, pp 63–70

Rostamian A, O'Hara JG (2022) Event prediction within directional change framework using a cnn-lstm model. Neural Computing and Applications 34(20):17193—17205

Salman O, Kampouridis M, Jarchi D (2022) Trading strategies optimization by genetic algorithm under the directional changes paradigm. In: 2022 IEEE Congress on Evolutionary Computation (CEC), IEEE, pp 1—8

Sharpe WF (1994) The sharpe ratio. Journal of portfolio management 21(1):49–58

Sun EW, Kruse T, Chen YT (2019) Stylized algorithmic trading: Satisfying the predictive near-term demand of liquidity. Annals of Operations Research 281(1):315–347

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

Vaidya R (2020) Moving average convergence-divergence (macd) trading rule: An application in nepalese stock market (nepse). Quantitative Economics and Management Studies 1(6):366–374

Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. Advances in neural information processing systems 30

Wu L, Wang Y, Wu LH (2022) Modeling index tracking portfolio based on stochastic dominance for stock selection. The Engineering Economist 67(3):172–194

Wu LC, Tsai IC (2014) Three fuzzy goal programming models for index portfolios. Journal of the Operational Research Society 65(8):1155–1169

Zeng X, Cai J, Liang C, et al (2023) Prediction of stock price movement using an improved nsga-ii-rf algorithm with a three-stage feature engineering process. Plos one 18(6):e0287754