Engineering Optimization

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/geno20

PASS: Portfolio Analysis of Selecting Strategies on quantitative trading via NSGA-II

Mu-En Wu, Ting-Chen Chen, Chien-Ping Chung, Guan-Rong Li, Da-Wei Chiang & Dong-Yuh Yang

ENGINEERING OPTIMIZATION
https://doi.org/10.1080/0305215X.2024.2418342

PASS: Portfolio Analysis of Selecting Strategies on quantitative trading via NSGA-II

Mu-En Wu[a], Ting-Chen Chen[b], Chien-Ping Chung[a], Guan-Rong Li[a], Da-Wei Chiang[a] and Dong-Yuh Yang[c]

[a]Department of Information and Finance Management, National Taipei University of Technology, Taipei, Taiwan; [b]Department of Quantitative Finance, National Tsing Hua University, Hsinchu, Taiwan; [c]Institute of Information and Decision Sciences, National Taipei University of Business, Taipei, Taiwan

ABSTRACT

Quantitative trading, driven by technological advancements, utilizes machine learning and mathematical models to develop trading strategies. Evaluating strategy performance through backtesting is common, but ensuring sufficient funding and stability is crucial. Portfolio construction mitigates risk and promotes sustained profit growth. This article introduces the portfolio analysis of selecting strategies (PASS) model, integrating a stability indicator, 'drawdown duration' (DDD) and multi-objective evolutionary algorithms (MOEAs). Results demonstrate that PASS reduces DDD risk by 73.9% and increases profit by 4.1% compared to other MOEAs. The model is applied successfully to trading markets like the New York Mercantile Exchange (NYMEX), the West Texas Intermediate (WTI) and the Mini Dow Jones Industrial Average (YM), aiding investors in crafting profitable strategies and managing risk in leveraged trades.

ARTICLE HISTORY
Received 22 March 2024
Accepted 14 October 2024

KEYWORDS
Quantitative trading; multi-objective optimization; Pareto optimality; genetic algorithm; drawdown

1. Introduction

Quantitative trading often involves finding market trends and constructing a portfolio to maximize returns. Effective portfolio construction requires expertise in finance, mathematics and computer science. A solid understanding of finance is crucial for trading, while mathematics aids in analysing profitability, and computer science helps optimize portfolios through evolutionary algorithms.

Traditionally, investors relied on moving averages, stochastic oscillators and price-based signals to detect market trends. Aguirre, Medina, and Duque (2020) used the moving average convergence divergence (MACD) indicator to find profitable strategies with genetic algorithms (GAs), and Wu, Syu, and Chen (2022) achieved outstanding performance in options.

In recent years, advances in machine learning have introduced more efficient and reliable methods for predicting market trends. In cryptocurrency trading, Fang *et al.* (2022) presented a comprehensive survey and indicated that there were promising opportunities for using machine learning technologies to create trading strategies. Vijh *et al.* (2020) employed artificial neural networks and random forest models to forecast the stock closing price for the next day, and Sebastião and Godinho (2021) adopted a machine learning strategy to profit even under adverse market conditions. Nabipour *et al.* (2020) used nine types of machine learning and two kinds of deep learning methods to predict stock trends. Meanwhile, Xiong *et al.* (2018) trained a deep reinforcement learning model that learned stock trading strategies and outperformed the Dow Jones Industrial Average. Additionally, Nayak and

CONTACT Dong-Yuh Yang yangdy@ntub.edu.tw

Misra (2018) used a GA to optimize parameters in a neural network to predict closing prices for the futures market.

In addition to predicting market trends, constructing a portfolio that balances profit and risk management was another critical aspect of trading. Therefore, Markowitz (1952) proposed the mean-variance theory, using two features to construct a portfolio by means of a mathematical framework. To address this challenge, multi-objective evolutionary algorithms have been widely used for the optimal portfolio. Fernandez *et al.* (2019) incorporated decision-maker preferences to find the optimal portfolio based on multi-objective evolutionary algorithms, and Chen, Lin, and Chen (2015) applied the grouping GA to the stock market to determine which stocks to buy and in what quantities.

Recent studies introduced novel algorithms and optimization methods to enhance trading strategies further. For example, Zhao, Wang, and Zhang (2019) showed significant promise in achieving global optimization by modelling the economic principles of supply and demand to guide the search process towards optimal solutions. Similarly, Shabani *et al.* (2020) mimicked the operations of search and rescue missions to explore the solution space efficiently and find optimal configurations, which is particularly useful in discrete variable problems. Li *et al.* (2020) presented the slime mould algorithm, a novel stochastic optimization technique inspired by the foraging behaviour of slime moulds, which effectively balanced exploration and exploitation to prevent becoming trapped in local optima. Sulaiman *et al.* (2020) took inspiration from the unique mating behaviours of barnacles to develop a bio-inspired algorithm that effectively handled complex engineering optimization problems by mimicking the natural selection and reproductive strategies of barnacles. Additionally, Moosavi and Bardsiri (2019) employed a human-behaviour-based approach that simulates the economic interactions between poor and rich populations to create a multi-population optimization framework, enhancing the diversity and convergence of the algorithm.

The above studies aimed to address the profitability problem, but they ignored the most important issue for investors, who hope to have more profit and not make a loss from their portfolio. Investors strongly valued the time of profit, which affected their mood and decision making. The fluctuation of investor emotions has increased the appeal of quantitative trading. The goal is to make quantitative trading more robust to overcome the impact of investor emotions on the time value of profits.

This study contributes to improving portfolio construction in two significant ways. First, a novel measurement index known as 'drawdown duration' (DDD) is introduced, designed to provide investors with a deeper understanding of their true earnings, moving beyond surface-level profit considerations in trading. Secondly, the portfolio analysis of selecting strategies (PASS) model is introduced to build portfolios composed of multiple strategies. The PASS model serves to counterbalance losses among strategies and amplify overall profits. However, it is essential to acknowledge that selecting the optimal combination of strategies for portfolio construction is a challenging NP-complete problem. The objective is to construct portfolios that balance profitability with risk mitigation, achievable through multi-objective optimization techniques like evolutionary algorithms (De Jong and Spears 1989).

The remainder of this research is structured as follows: Section 2 provides a brief introduction to the evolutionary algorithm and the performance indicators evaluated, ensuring the article's self-containment. Section 3 defines and elucidates the PASS model, detailing the basic algorithm and the DDD indicator. Section 4 presents the quantitative trading strategies using the PASS model and reports the experimental results across different futures markets, evolutionary algorithms and hyperparameters. Finally, Section 5 wraps up the article and suggests potential directions for future research.

2. Related work

This section focuses on addressing investment risk, a key concern for investors, through the use of evolutionary algorithms and various risk indicators. Numerous risk measures, such as profit, Sharpe ratio, value-at-risk and maximum drawdown (MDD), have been developed to express risk comprehensively, while evolutionary algorithms, including GAs, particle swarm optimization and artificial bee colony algorithms, have been applied to optimize investment strategies. Special attention is given to multi-objective optimization techniques, particularly NSGA-II, where non-dominated sorting and crowding distance play a critical role in solving multi-objective problems. This section also emphasizes the importance of backtesting in evaluating the performance of investment strategies using these indicators.

2.1. Genetic algorithms

The GA, introduced by Holland (1992), is a popular optimization technique inspired by natural selection and evolution. It utilizes a 'survival of the fittest' approach to solve complex optimization problems. In the GA, potential solutions were represented as chromosomes, and the algorithm processed them using operations like crossover, mutation and selection to generate better offspring. The most effective solutions, based on their fitness score, were retained to create the next generation of chromosomes. The iterative process continued until the algorithm converged and identified the best solution.

The application of multi-objective optimization expanded in various fields, leading to the development of numerous algorithms and approaches. For instance, the optimization method of passive omnidirectional buoy arrays in on-call anti-submarine search was enhanced by incorporating an improved NSGA-II. Bi *et al.* (2024) considered search probability, effective positioning probability, and delivery amount as objective functions, establishing a robust framework for optimizing sonar buoy deployment. Similarly, the multi-objective Bayesian optimization (MOBOpt) algorithm introduced a novel approach to calculate Pareto front approximations with fewer evaluations, proving effective for costly objectives (Galuzio *et al.* 2020).

While GAs have been effective in solving various optimization problems, they were primarily suited for single-objective optimization. However, many quantitative trading problems involved multiple objectives that need to be optimized concurrently. Although it was feasible to apply GAs to multi-objective optimization, researchers faced several challenges in designing a suitable approach.

In the maritime sector, an improved non-dominated sorting genetic algorithm (NSGA-III) was utilized to optimize ship weather routing, considering operational costs, sailing time and CO_2 emissions. The enhanced algorithm addressed convergence difficulties by integrating a route from sampling with the A$*$ (A-star) algorithm, showcasing significant reductions in ship operating costs and emissions (Ma *et al.* 2024). Moreover, the multi-objective optimization of the environmental–economic dispatch problem was advanced by a reinforcement learning-based NSGA-II, demonstrating superior results in balancing conflicting objectives and achieving a well-distributed Pareto front (Bora, Mariani, and dos Santos Coelho 2019).

In conclusion, GAs such as NSGA-II (Deb *et al.* 2002) and other multi-objective evolutionary algorithms (MOEAs) like ant colony optimization (Dorigo, Maniezzo, and Colorni 1996), the artificial bee colony algorithm (Karaboga 2005) and particle swarm optimization (Kennedy and Eberhart 1995), offered a viable solution for addressing a variety of optimization problems. Researchers needed to select the appropriate approach and parameter settings judiciously to achieve optimal outcomes. While GAs had limitations, such as their incapacity to handle multiple objectives and constraints, they remained a valuable tool in the realm of optimization problem solving.

2.2. The non-dominated sorting genetic algorithm

NSGA-II, introduced by Deb *et al.* (2002), was an improvement over GAs that incorporated non-dominated sorting (Srinivas and Deb 1994) to achieve Pareto optimality. The algorithm selected superior chromosomes through elitism, followed by ranking based on non-dominated sorting and crowding distance. The process of NSGA-II is illustrated in Figure 1.

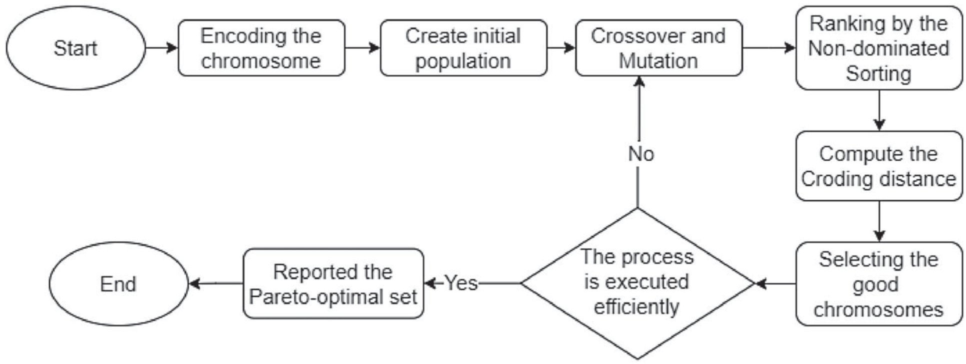


Figure 1. Flowchart of the NSGA-II.

The optimization of double pipe heat exchangers benefited from evolutionary multi-objective approaches, with the non-dominated sorting genetic algorithm with reinforcement learning (NSGA-RL) outperforming traditional methods like NSGA-II and the chaotic non-dominated sorting genetic algorithm (CNSGA). The improvement was evident in the significant enhancements in thermal performance indices and heat transfer rates achieved through optimal design configurations (Colaço *et al.* 2022). Furthermore, a parallel NSGA-III tailored for multi-period portfolio optimization was developed, merging the strengths of parallel genetic algorithms (PGAs) and NSGA-III to enhance risk management and return optimization in real-world investment scenarios (Qian and Wang 2024).

NSGA-II was widely recognized as a leading multi-objective evolutionary algorithm (MOEA) owing to its effectiveness in addressing complex real-world multi-objective optimization problems (Gunantara 2018). Applications of multi-objective optimization included portfolio risk, profit and trading cost optimization (Meghwani and Thakur 2018), as well as the evolution of multi-modal robot behaviour (Huizinga and Clune 2022). Additionally, it was applied to solve multi-commodity capacitated network design problems (Kleeman *et al.* 2012). Hadka and Reed (2012) conducted a comparative study of many-objective evolutionary optimization against search controls and failure modes.

NSGA-II integrates non-dominated sorting and crowding distance, two key techniques derived from GAs. Non-dominated sorting categorizes chromosomes into different levels of non-domination, with chromosomes selected from the top level until the desired number is attained. In this context, non-domination is defined for a set of solutions S, where a solution i is deemed to dominate solution j if it satisfies two criteria. First, solution i must be no worse than solution j on all objectives, indicating at least equivalent performance. Secondly, solution i must outperform solution j on at least one objective, demonstrating superiority. Thus, solution i is considered a better option than solution j.

Crowding distance is employed by NSGA-II to rank chromosomes at the same non-domination level, thus preventing local optima and facilitating the exploration of optimal solutions. The computation of crowding distance involves calculating the distance for each solution. The formula for the distance of the solution is

$$dis_M(i) = \frac{S_M(i+1) - S_M(i-1)}{S_{M,\max} - S_{M,\min}}, \tag{1}$$

where M is each objective, $S_M(i)$ is the value of objective M for the solution i, $i-1$ and $i+1$ are the previous and the next solutions, $S_{M,\max}$ is the maximum score of the objective and $S_{M,\min}$ is the minimum score of the objective. The crowding distance denoted by $C\,dis(i)$ for a solution i is the sum of distances for all objectives, using the following formula:

$$C\,dis(i) = \sum_{M=1}^{n} dis_M(i). \tag{2}$$

If the crowding distance does not increase for subsequent iterations, the algorithm returns the chromosomes as the Pareto-optimal solution set; otherwise, it goes back to the crossover step and repeats the following steps.

2.3. The indicator of evaluated performance

Investors commonly use various indicators to assess the performance of investment strategies through backtesting. These indicators encompass profit, profit factor (Daskalakis and Markellos 2008), Sharpe ratio, win rate, value-at-risk (VaR) and Drawdown (Magdon-Ismail *et al.* 2004), among others. The primary objective for investors is to maximize profit while minimizing losses and overall risk.

The Sharpe ratio, which quantifies the excess return per unit of risk, underwent refinements by researchers like Israelsen (2005), who adjusted its denominator to reflect excess returns more accurately. The trend ratio (Chou *et al.* 2021), which incorporated investor psychology into the Sharpe ratio, was also employed in portfolio optimization (Chou, Jiang, and Kuo 2021). Tsang and Chen (2018), for instance, employed directional change indicators in the foreign exchange market.

Another popular risk measure is VaR, which quantifies the risk of potential losses with certain confidence (Lin and Ko 2009). Bradshaw *et al.* (2009) designed a risk criterion based on VaR to optimize the portfolio, while Glasserman, Heidelberger, and Shahabuddin (2002) used VaR and surplus variance to maximize the expected return of the portfolio, both using evolutionary algorithms for optimization.

Apart from profit-based risk measures, drawdown is a pure risk indicator that quantifies losses by measuring the decline from the strategy's current record high. A decrease in profit can negatively impact an investor's confidence. MDD represents the maximum decrease in accumulated profit magnitude, indicating how much percentage of assets could potentially be lost in the worst-case scenario. Drenovak *et al.* (2022) proposed an approach based on MDD to optimize buy-and-hold portfolios, while Choi (2021) applied MDD to predict asset prices using stock selection rules.

3. Modelling risk in portfolio optimization

This section addresses the fundamental research question by developing a model that provides deeper insights into trading earnings, moving beyond surface-level profit assessments commonly used in quantitative trading. The process is illustrated in Figure 2. The process begins by defining the research question and initializing the model. Then, the DDD definitions are presented, followed by a discussion of the model's time complexity. Subsequently, two objectives, namely profit and DDD, are optimized to construct a portfolio using a multi-objective efficient search algorithm. Walk forward analysis is used to simulate actual trading.

3.1. Problem statement

In high-risk quantitative trading, investors often face the challenge of significant losses and long recovery times, making robust risk management strategies essential. Recognizing the need for robust risk management strategies, the pivotal question emerges: 'How can significant losses inherent in quantitative trading be effectively addressed, recovery time minimized, and portfolio reliability bolstered through the integration of multiple strategies?'

To address this question, the model uses a strategy pool composed of multiple trading strategy prototypes (TSPs), each with different parameters. The trader generates the strategy pool from indicators,

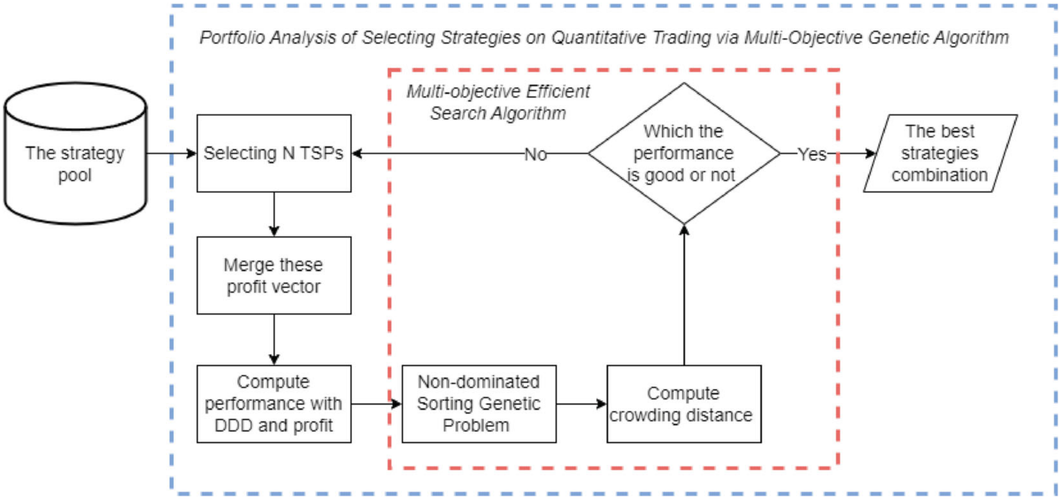


Figure 2. Flowchart of the PASS model.

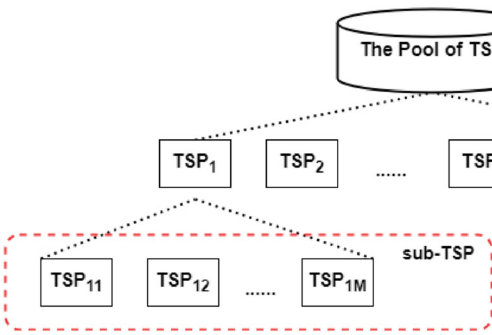


Figure 3. The architecture of the TSP.

Table 1. Profit of the trading date.

Trading date	Profit/loss
date A	−3,000
date B	21,400
date C	102,000
date D	−3,000
date E	32,200
date F	−3,000

methods and sign quantitative trading, including trend, counter-trend, swing and day trading, as shown in Figure 3. Each TSP is a specific trading strategy and contains branch sub-TSPs with different parameters and is represented as TSP_{11}, TSP_{12},..., TSP_{1n},..., TSP_{1N}. Additionally, no two TSPs have the same performance.

For example, the relative strength index (RSI) strategy includes the γ period and the α and β levels for the oversold and overbought conditions. For the RSI strategy, if the intervals of each parameter are set to $\gamma \in [10, 100]$, $\alpha \in [0, 50]$, $\beta \in [51, 100]$, the strategy pool has $90 \times 50 \times 50 = 225{,}000$ kinds of sub-TSPs. All TSP performances are computed to generate the profit of the trading date, as shown in Table 1.

Table 2. Unrealized and realized profit of trading date.

Trading date	Profit/loss	Status of profit
timestamp 1	−3,000	Realized
timestamp 2	−3,800	Unrealized
timestamp 3	−2,500	Unrealized
timestamp 4	−1,200	Unrealized
timestamp 5	−800	Unrealized
timestamp 6	21,400	Realized
timestamp 7	22,500	Unrealized
timestamp 8	22,000	Unrealized
timestamp 9	23,750	Unrealized
timestamp 10	25,750	Unrealized
timestamp 11	10,200	Realized

The conventional evaluation of portfolio profitability, which focuses primarily on realized profits, has a critical limitation: it overlooks the potential for sustained profitability and the latent unrealized profits within the portfolio. By fixating solely on these straightforward profit measures, the portfolio's potential for prolonged and consistent profitability—a crucial consideration for investors—is inadvertently sidestepped. Furthermore, the approach disregards the latent, unrealized profit potential residing within the portfolio, failing to account for its capacity to deliver continuous returns. To address these shortcomings and offer a more comprehensive assessment, it is imperative to evaluate both realized and unrealized profits concurrently. A thorough daily analysis of profits generated by each strategy is necessary to represent the portfolio's true profitability accurately, as vividly depicted in Table 2.

In light of the aforementioned challenges, the PASS model was developed to enhance portfolio performance. The PASS model integrates TSPs from a strategy pool, aiming to maximize profit while accurately measuring both realized and unrealized profits and losses during the backtesting period. To optimize the portfolio within the PASS model, two objective functions are formulated. The first objective function seeks to maximize the profit (P) of the portfolio, which is represented by the total sum of profits generated by the selected TSP_i. The profit of each TSP_i can be expressed as follows:

$$P_i = \sum_{\substack{i=1,2,\ldots,N \\ j=1,2,\ldots,M}} P_{i_j}. \tag{3}$$

The objective function can be expressed as follows:

$$\text{Maximize } P = \arg\max_i \; P_i. \tag{4}$$

In this context, N represents the number of strategies in the portfolio, and S, DDD, and D are subject to non-negative constraints. P_{i_j} represents the profit of a specific sub-strategy in the ith TSP, TSP_i.

The second objective function aims to enhance daily profits continuously beyond previous thresholds. Managing the maxDDD is key to minimizing its impact on the portfolio's overall performance The definition of maxDDD is

$$\mathrm{maxDDD} = \max_{i=1,2,\ldots,N} (\mathrm{DDD}_i). \tag{5}$$

The objective function can be concisely expressed as: minimize D, where D is defined as the maximum of the values of DDD.

$$\text{Minimize } D = \min(\mathrm{maxDDD}). \tag{6}$$

Here, D represents maxDDD, while DDD represents the drawdown duration for a specific subset of TSP_i within the portfolio. Through the implementation of these objective functions, the PASS model

Table 3. Parameters used in multi-objective GA analysis.

Parameter	Description
Population size	Number of chromosomes in the population
Chromosome length	Number of genes in each chromosome
Crossover rate	Probability of crossover occurring between two chromosomes
Mutation rate	Probability of a gene mutating within a chromosome
Number of generations	Total number of generations for evolution
Elite strategy	Number of top chromosomes retained in each generation
Objective 1	Profit
Objective 2	maxDDD

facilitates the construction of portfolios that maximize profit while effectively managing potential losses during the backtesting period. In the subsequent sections, the methodology, algorithms, experimental setup, data sources and performance evaluation metrics employed to validate the effectiveness of the proposed approach will be elaborated.

For the work of PASS, the following steps are followed: the first step, called the backtest step, involves computing the unrealized/realized profit of the trading date to explain implied volatility. In the second step, called the select step, several TSPs' unrealized/realized profit of the trading date are selected and merged to form a portfolio from the strategy pool, and these are input for subsequent strategy selections. In the third step, called the MOEG (multi-object evaluation genetic) step, the non-dominated level and crowding-distance performance are calculated by maxDDD and profit in the NSGA-II. In the final step, known as the assess step, the portfolio's performance is evaluated. If the portfolio is deemed optimized, the best TSP number is recorded; otherwise, the process returns to the second step for further optimization. The parameters implemented in the multi-objective GA analysis, as listed in Table 3, include population size, chromosome length, crossover rate, mutation rate, number of generations and the elite strategy. These parameters are crucial for optimizing the selection of TSPs in the PASS model.

3.2. DDD definition and time complexity of model

When evaluating trading strategy performance, traders often rely on metrics derived from actual trades, such as the Sharpe ratio and profit factor. However, these indicators often leave a critical aspect unexamined: the true extent of profit accrued. Although metrics like drawdown and profit factor help assess portfolio performance, they fail to capture the remaining profit potential and the time required to generate additional profits. Additionally, backtesting alone may not fully reveal the risks of a strategy, making it difficult for traders to assess its effectiveness accurately.

To address these limitations, a novel risk indicator called DDD is introduced to provide a more comprehensive view of trading strategy performance. The indicator represents a departure from the conventional drawdown (DD) metric, aiming to provide a holistic solution that encompasses sustained profit growth and effective risk mitigation. The formula for DDD is rooted in the fundamentals of drawdown. To calculate the DDD indicator, let t be a timestamp with $t \in \{1, 2, \ldots, T\}$, T be the latest time point, $P(t)$ be the unrealized/realized profit of the trading date at timestamp t, $Cum.P(t)$ be the accumulated profit and loss at timestamp t and $Cum.P_{\max}(t)$ be the highest accumulated profit of t. The formula for DD is as follows:

$$\mathrm{DD}(t) = Cum.P_{\max}(t) - Cum.P(t), \tag{7}$$

where $\mathrm{DD}(t) = 0$ is the previous record-high broken at timestamp t, $i \in \{1, 2, \ldots, T\}$, T being the latest timestamp of $\mathrm{DD}(t) = 0$, and DDD is defined as follows:

$$\mathrm{DDD} = \left\{ \frac{(h_i - h_{i-1} - 1)}{T} \,\middle|\, i = 2, \ldots, T \right\}, \tag{8}$$

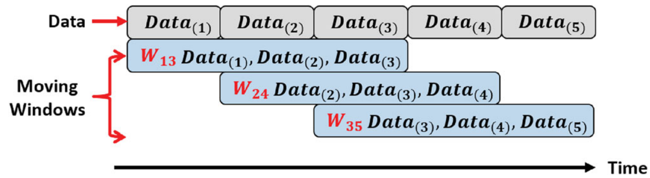


Figure 4. The moving window about strategy.

where DDD is the ratio that displays how many dates the strategy broke the previous record-high, and h is one element of the set H that records each date when the previous record-high was broken, defined as follows:

$$H = \{h_1, \ldots, h_n \mid \text{DD}(h_i) = 0,\ \forall\, i, j \in 1, 2, \ldots, T,\ h_i < h_j,\ \forall\, i < j\}. \tag{9}$$

In this research, maxDDD is introduced as the maximum element of DDD, measuring the maximum duration required to surpass previous record-high profitability levels. Consequently, traders gain valuable knowledge about how long a strategy may take to generate additional profits. For example, if a trading strategy's accumulated profit on each date has $P = \{10, 20, 10, 15, 20, 30, 25, 50, 30, 35, 60\}$ and $H = \{10, 20, 30, 50, 60\}$, with $N = 11$, the calculated DDDs are DDD_1 $= [(20 - 10) - 1]/11 = 0.81$, DDD_2 $= [(30 - 20) - 1]/11 = 0.81$, DDD_3 $= [(50 - 30) - 1]/11 = 1.73$ and DDD_4 $= [(60 - 50) - 1]/11 = 0.81$. Therefore, maxDDD $= \max\{0.81, 1.73\}$, which equals 1.73, meaning that breaking the previous record high required 1.73 backtesting periods. However, if the length of the strategy pool matches the backtesting period, the division of the backtesting period becomes inconsequential. In Section 4.2, the correlation between maxDDD and MDD is displayed through the experiment when maxDDD is used.

To test whether a portfolio is overfitting, the walk forward analysis (WFA) method is utilized (Pardo 2012). Since the historical data of trading is a time series, near-current data are highly correlated. WFA segments the historical data and combines several historical data to define windows and backtesting periods. The windows are defined by the variable Δ, which ranges from 1 to T. A backtesting period, W_{ij}, is from $Data_i$ to $Data_j$, where $Data$ represents the original profit based on the timestamp, and T is the latest timestamp, W is as follows:

$$W_{ij} = \{Data_i, Data_{i+1}, \ldots, Data_j \mid i = 1, 2, \ldots, T - \Delta + 1, j = \Delta, \Delta + 1, \ldots, T\}. \tag{10}$$

For example, if T were 5, and the data were segmented with $\Delta = 3$, then W is represented as $\{W_{13}, W_{24}, W_{35}\}$. Specifically, $W_{13} = \{Data_{(1)}, Data_{(2)}, Data_{(3)}\}$, $W_{24} = \{Data_{(2)}, Data_{(3)}, Data_{(4)}\}$, $W_{35} = \{Data_{(3)}, Data_{(4)}, Data_{(5)}\}$, as illustrated in the Figure 4.

Hence, one straightforward approach is to use brute force to search for the best strategy portfolios with the WFA. The process is outlined in Algorithm 1.

To analyse the time complexity of Algorithm 1 in the context of the WFA, Big-O notation is applied. Each window in the analysis contains the same $\Delta^* Data(i)$, resulting in a time complexity of $\Delta \times B$, where $B = O(1)$ represents the base of backtesting data. The backtesting updates, considering the trading strategy pool TSP with N_{tsp} strategies inside, demonstrate a time complexity of $N \times \Delta \times B$.

During the WFA process, $T - \Delta + 1$ windows are generated, each containing C_m^n portfolios, indicating the selection of m from n strategies. These portfolios require E evaluations for performance assessment. The time complexity of computing portfolio performance within each window is illustrated in Figure 5.

The overall computational complexity is determined by the average computational complexity of the operation process, which can be expressed as a function of Δ, T and E:

$$X_{\text{Average}} = O\Big((T - \Delta + 1)(N_{\text{tsp}} \times \Delta B + C_n^m \times E)\Big). \tag{11}$$

Algorithm 1 Brute force backtesting algorithm

Require: *Windows*- - -a list of time windows to be tested; *strategy_pool*- - -a pool of trading strategies includes various TSPs; *Data*- - -a dataset containing historical market data

Ensure: *TSP_records*- - -a list of TSPs contains two indicators: DDD and profit

```
for W_ij in Windows do                                              ▷ Loop over all time windows
    records = []
    for TSP in strategy_pool do
        TSP_records = []          ▷ Initialize empty list of trading records for the current TSP
        for data in Data_i to Data_j do
            Append TSP's trading record in data into TSP_records  ▷ Apply strategy to data and store trading record
        end for
    end for
    Append TSP_records into records
    Compute DDD and profit indicators by TSP_records
end for
```

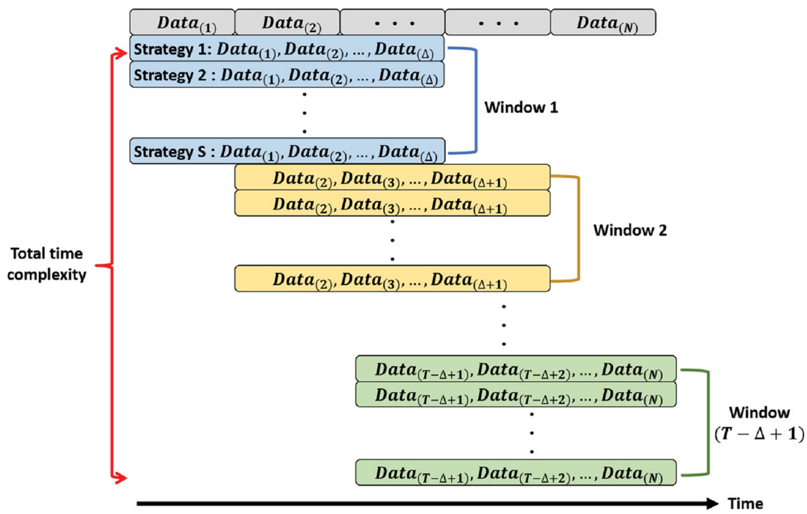


Figure 5. The time complexity of each strategy portfolio performance in the window.

When WFA is not employed ($\Delta = T$), the computational complexity follows a specific pattern, as shown in the following equation:

$$X_{\text{Best}} = O\Big(N_{\text{tsp}} \times T \times B + C_n^m \times E\Big). \tag{12}$$

Conversely, considering each data point as an individual window ($\Delta = 1$) results in a different computational complexity, as shown in the following equation:

$$X_{\text{Worst}} = O\Big(T(N_{\text{tsp}} \times T \times B + C_n^n \times E)\Big). \tag{13}$$

As shown in Figure 5, the brute force method has heavy computation, making it impractical for large-scale datasets. To address this, an efficient search algorithm is proposed in the following subsection, which significantly improves the computational complexity and optimization process of the strategy portfolio.

3.3. Multi-objective efficient search algorithm with portfolio optimization

This subsection outlines the multi-objective efficient search algorithm for portfolio optimization, focusing on the PASS model, which aims to create a stable and profitable strategy pool. The model selects n profitable TSPs to generate a profitable strategy and incorporates the maxDDD metric to assess the performance stability of the TSP combination. To resolve conflicts between multiple objectives, NSGA-II with Pareto fronts is employed to enhance the efficiency of the strategy search process. Other multi-objective efficient search algorithms are also explored to identify the most suitable one for the model, as discussed in Section 4.2.

In the preparation phase of NSGA-II, chromosomes are defined as feasible solutions to the optimization problem. Each chromosome consists of multiple genes, with each gene representing a specific TSP. Initially, the number of genes and the chromosome size are determined by selecting the five best-performing chromosomes from a pool of two thousand strategies, with each chromosome containing three genes. After setting the hyperparameters, the initialization step randomly selects solutions.

Next, genetic operators are applied, during which genes are randomly exchanged between two chromosomes through crossover. The 'Crossover Rate' hyperparameter governs the eligibility of chromosomes for crossover. After the crossover process, mutations are introduced to maintain chromosome diversity and allow exploration beyond local optima. Genes within each chromosome are mutated according to a predetermined mutation rate, where a selected chromosome has one of its genes replaced with a randomly chosen alternative. Moreover, the approach implements an 'Elite Strategy' to preserve the most outstanding chromosomes in each iteration.

Finally, each chromosome's performance is evaluated by comparing it with other TSPs, following the process outlined in Algorithm 2. A TSP is defined as having two items: a set S indicating which chromosomes are dominated by the current chromosome, and another, n, representing the number of chromosomes that dominate the current one. The comparison function, denoted as '$>_{Perf.}$', evaluates maxDDD and profit. If true, the TSP shows an advantage over another.

Algorithm 2 provides a detailed description of the Pareto front ranking algorithm. In essence, the algorithm compares each chromosome based on its performance with respect to the two objectives (profit and stability). If a chromosome outperforms another chromosome in both objectives, it dominates the other chromosome, and its ID is added to set S. Conversely, if a chromosome is outperformed by another chromosome in either objective, its n value is incremented by one.

To formalize the set S for each chromosome i, the following definition is used:

$$S_i = \{j \mid P_i > P_j \text{ and } D_i \leq D_j\}, \tag{14}$$

where $i \in \{1, 2, \ldots, n\}$ represents the ID of the chromosome, $j \in \{1, 2, \ldots, n\}$, $j \neq i$, represents the ID of another chromosome (excluding i), S_i denotes the set of chromosomes dominated by i, P_i represents the profit of chromosome i, D_i denotes the maximum distance deviation of chromosome i, and $>_{Perf.}$ indicates one chromosome being better than another chromosome. Chromosomes are selected starting from the smallest n, incrementing in subsequent iterations until a sufficient number have been chosen.

Figure 6 presents Algorithm 2 for selecting the best chromosomes based on two objectives: profit and stability. Each dot represents a chromosome with a different TSP combination. Specifically, the hypothesis is that higher profit would lead to higher stability, and higher stability would lead to higher profit.

For example, in Table 4, chromosome A emerges as the superior chromosome based on the Pareto front. It dominates other chromosomes and is not dominated by any other. It is important to note that chromosomes B, C and D do not dominate each other, as each possesses its own strengths. Conversely, chromosome E is the weakest and is dominated by others. All the chromosomes in the table can be computed and ranked accordingly.

Algorithm 2 Pareto front ranking

```
Require: strategy_pool- - -a pool of trading strategies includes various TSPs
Ensure: Rank- - -a TSP's list of Pareto fronts ranking
for TSP in Strategy_pool do
    TSP.S = []                                    ▷ Initialization TSP ranking.
    TSP.n = 0
    for other_TSP in Strategy_pool do
        if TSP >_Perf. other_TSP then
            Append other_TSP into S
                                                  ▷ Other_TSP dominated by TSP
        end if
        if other_TSP >_Perf. TSP then
            TSP.n = TSP.n + 1                     ▷ TSP dominated by Other_TSP
        end if
    end for
end for                                           ▷ Compute TSP ranking
Rank = []
i = 0
Remain = Strategy_pool
while Remain.empty() == False do
    Temp = []                                     ▷ Not currently calculated for ranking
    for TSP in Remain do
        if TSP.n == 0 then
            Append TSP into Rank[i]
        else
            TSP.n = TSP.n − 1
            Append TSP into Temp
        end if
    end for
    Remain = Temp
    i = i + 1
end while
```

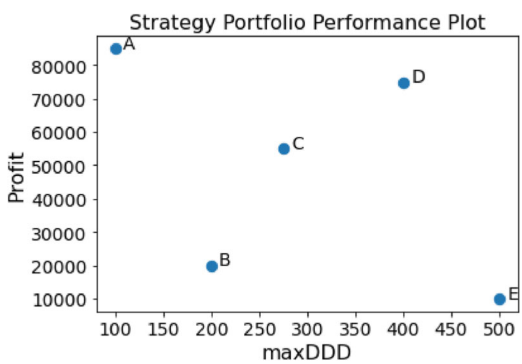


Figure 6. The dot chart of two indicators.

Thus, the process commences with chromosome A at level 1 of the Pareto front. If level 1 of the Pareto front is empty, n is decremented in all chromosomes, and the chromosome with n equal to zero is selected for the next iteration, as shown in Figure 7.

Table 4. The iterations for the ultimate moving average.

Chromosome	n	S
A	0	B, C, D, E
B	1	E
C	1	E
D	1	E
E	4	Ø

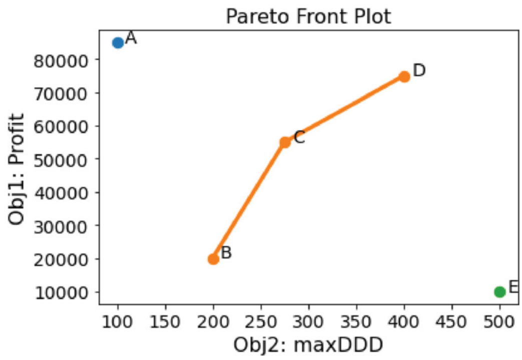


Figure 7. Pareto front plot.

To proceed with the selection of superior and more diverse chromosomes at the same level for subsequent iterations, the distance ranking is determined by calculating the Euclidean distance between each chromosome and its two nearest neighbours on the same Pareto front level. The chromosome selected at the same level is then chosen based on the computed distance ranking using Algorithm 3.

Algorithm 3 Distance ranking

Require: *maxDDD*- - -an indicator of evaluator, *profit*- - -an indicator of evaluator, *Rank*- - -a TSP's list of Pareto fronts
Ensure: *distance_list*- - -a list of rankings based on crowding distance, with each ranking corresponding to a TSP

$Object = [maxDDD, profit]$
for obj in $Object$ **do**
  for i in 0 to $rank.size()$ **do**
    for j in 1 to $rank[i].size()$ **do**
      $distance = (rank[i][j+1] - rank[j-1])/_{obj}$ $(\mathbf{max}(Strategy\_pool) - \mathbf{min}(Strategy\_pool))$
      Append *distance* into *distance_list*
    end for
  end for
end for

The selection rules for subsequent iterations are as follows.

- The chromosome from the top level of the Pareto front is chosen.
- If chromosomes are at the same level of the Pareto front, chromosomes with a more significant distance are prioritized by distance ranking.

Overall, the PASS model can identify the most beneficial strategies in the strategy pool using two measurements, DDD and profit, to reduce risk and increase profit. The multi-objective efficient search algorithm helps reduce time complexity and achieves results comparable to brute force search. It provides traders and researchers with a quick way to find stable and profitable strategies.

4. Experimental results

In this section, an experimental evaluation is conducted within the PASS model. Chromosomes are defined through a combination of strategies, with two examples being the 'momentum breakout strategy' (MBS) and the 'ultimate moving average strategy' (UMAS) (Chan 2009). With appropriate hyperparameters, these strategies are input into the PASS model, and their performance is evaluated using accumulated profit charts and MDD-DDD scatter charts. Accumulated profit charts offer insights into loss volatility and profit growth smoothness, while MDD-DDD scatter charts illustrate the convergence of DDD across iterations. Additionally, a thorough hyperparameter analysis is performed to identify the optimal configuration for the PASS model, enabling optimization of its performance.

4.1. Quantitative trading strategy

The quantitative trading strategies employed in this research are presented. The first strategy is called the 'momentum breakout strategy' (MBS). *Amplitude* is defined as the difference between the highest price before the current day ($H_{\text{daily}}[-1]$) and the lowest price before the current day ($L_{\text{daily}}[-1]$), observed on the previous trading day. The amplitude is computed using the following equation:

$$Amplitude = H_{\text{daily}}[-1] - L_{\text{daily}}[-1]. \tag{15}$$

To capture the momentum effect, the *momentum* is calculated by multiplying the amplitude with a *Coefficient*, which ranges from zero to one.

$$momentum = Amplitude \times Coefficient. \tag{16}$$

Additionally, the *momentum* and daily opening price (O_{daily}) are applied to determine the long position using the equation

$$call = O_{\text{daily}} + momentum, \tag{17}$$

and the *stop loss* mechanism to determine the price at which a position should be closed. The stop loss price (*stop loss*) is computed by multiplying the call position price (*call*) with a coefficient (β), which reflects the acceptable level of loss (ranging from 0.7 to 0.95):

$$stop\ loss = call \times \beta. \tag{18}$$

After calculating the MBS indicator, a long position is initiated at the *call* price if the highest price within a minute ($H_{1\,\text{min}}$) exceeds the *call* price. Once a position is opened, the lowest price within a minute ($L_{1\,\text{min}}$) is monitored. If the lowest price falls below the *stop loss*, the position is closed using the *stop loss*. Otherwise, the position is closed at the end of the daily trading block. For the TXF instrument, transfer tax, handling fees, and slippage are accounted for by deducting five points from the recorded profit.

The second strategy is the 'ultimate moving average strategy' (UMAS), which defines the *call* indicator as follows:

$$call = kMA_{30\,\text{min}}[0] > kMA_{30\,\text{min}}[-2], \tag{19}$$

where k represents an integer value ranging from 10 to 90 with a step size of 10, kMA denotes the average of the last k closest prices, $t = [0]$ represents the current time and $t = [-2]$ represents the time before 60 minutes. Additionally, a fixed percentage of pyramiding, referred to as *pyramiding*, is added to the price of the subsequent long position using the following formula:

$$pyramiding = \text{the previous price of the call position} \times \Delta, \tag{20}$$

where Δ is a floating-point value range from 0.001 to 0.01 with a step size of 0.0005. If $pyramiding \geq C_{1\,\text{min}}$, a long position is opened at the price of *pyramiding*, and the next price of *pyramiding* is updated. Otherwise, all positions are closed under two conditions: when the last closest price of the one-minute candlestick is lower than the previous price of the long position, or when the futures closing date is reached.

4.2. The feasibility of PASS

This subsection rigorously applies the strategies outlined earlier to the PASS model for a thorough evaluation of their effectiveness. The evaluation spans a comprehensive backtesting period from January 2017 to July 2021. The strategies' efficacy and performance are systematically quantified and visually represented through two key graphical tools: the accumulated profit chart and the MDD-DDD scatter chart.

The accumulated profit chart visually depicts a smooth trajectory of profit growth, accompanied by diminishing losses. The chart provides valuable insights into the performance dynamics of the strategies over time, offering a clear picture of their consistency and potential for sustained profitability.

Meanwhile, the MDD-DDD scatter chart offers a clear depiction of the complex relationship between the MDD and its corresponding DDD across successive iterations. Designed to be user-friendly, it offers distinct advantages when setting the MDD value to be negative, indicating a convergence process toward the upper-left quadrant. By closely examining the upward-left movement of data points within this chart, a deep understanding can be gained of how the chromosome configurations effectively mitigate DDDs.

In conducting the experiments, specific hyperparameters of NSGA-II were utilized, with the TSP as the gene and the portfolio as the chromosome. These hyperparameters included the initial number of chromosomes, the number of genes, the number of iterations, the crossover rate and the mutation rate. The initial number of chromosomes was set to 500, the number of genes to 3, the number of iterations to 100, the crossover rate to 0.8 and the mutation rate to 0.1.

The PASS model execute time required for the analysis primarily depends on the duration of evaluating each strategy. Specifically, calculating the daily profits for each strategy necessitates converting trading records into daily profit data. It takes approximately two hours to compute the daily profits for 2600 TSP instances, and an additional 200 seconds to select the best chromosome.

Upon completion, the PASS model generated a series of portfolios as recommendations. Investors can choose between a more profitable portfolio with higher risk or a less profitable portfolio with lower risk, depending on their risk appetite. To ensure a balanced portrayal, the median portfolio was chosen for visualization purposes. Figure 8 presents portfolios from the previous strategy at level 1 on the Pareto front.

In Figure 9, the accumulated profit chart is presented, depicting the performance of the MBS within the PASS model. Upon analysing the results, it was observed that the best chromosome consisted of two instances of gene 425 and gene 426. Despite optimizing the MBS through the PASS model, no significant improvement was observed, as the best chromosome still comprised similar genes. This finding suggests that the backtesting results did not yield notable enhancements for this particular strategy.

In contrast, Figure 10 showcases the accumulated profit chart that represents the optimized performance of the UMAS within the PASS model. The best chromosome configuration identified in

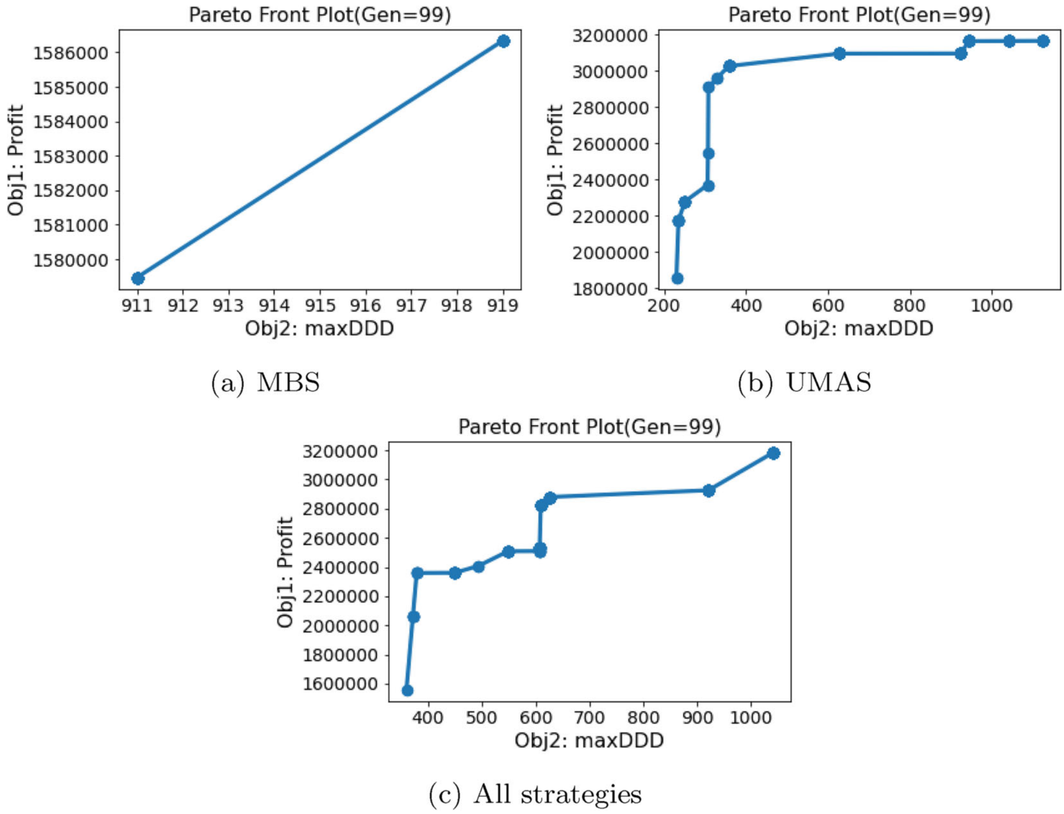


Figure 8. The portfolios at level 1 on the Pareto front. (a) MBS. (b) UMAS and (c) All strategies.

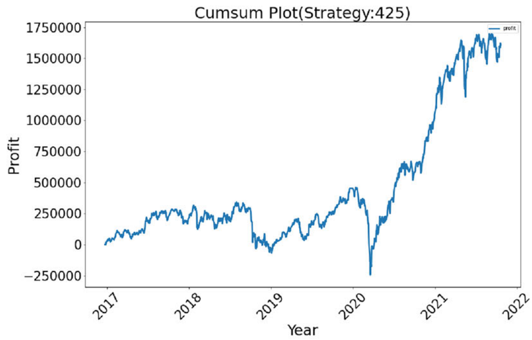
this case included gene 2167, gene 2249 and gene 2131. Remarkably, the configuration demonstrated a significant improvement in mitigating drawdowns, as evidenced by the two circles highlighting the performance of gene 2167, gene 2249 and gene 2131. The accumulated profit line of the chromosome exhibited a steady rise, indicating a notable enhancement in the strategy's performance.

Furthermore, Figure 11 displays the accumulated profit chart for all strategies implemented in the PASS. The best chromosome configuration, comprising gene 2246, gene 2172 and gene 2128, exhibits notable improvements in mitigating drawdowns. The circles labelling genes 2246, 2172 and 2128 illustrate their contributions to the steady rise in the accumulated profit line. Surprisingly, these chromosomes belong to the UMAS, indicating its superior suitability compared to the MBS.

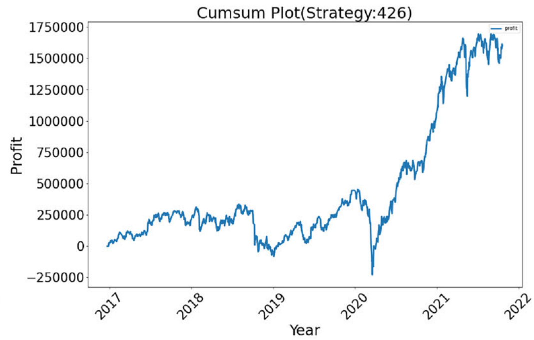
Based on the experimental results, a smooth and steady rise in the accumulated profit line of the chromosome is observed. Moving forward, investors aim to enhance the MDD and DDD further. Additionally, the convergence of the best chromosome's DDD and MDD is examined through several iterations, providing valuable insights for improvement.

In Figure 12, the iterations for the UMAS are showcased, demonstrating the convergence of chromosomes towards the upper left quadrant of the scatter chart. Notably, the circles highlight the most significant changes from the first to the ninth iteration, suggesting a marked improvement in the strategy's performance.

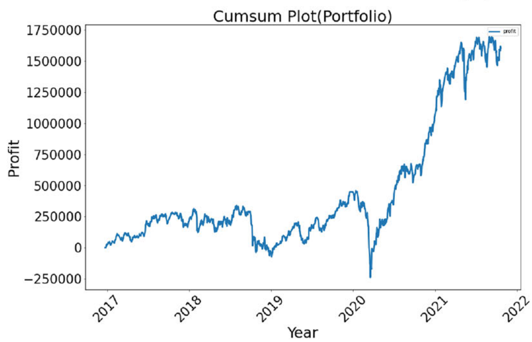
Additionally, Figure 13 highlights the most notable changes across all strategies, particularly between the first and twelfth iterations, resulting in a reduced DDD, as indicated by the circle in the upper left quadrant. Subsequently, the chromosomes gradually converge towards the upper left quadrant, demonstrating promising progress until the seventeenth iteration.



(a) Gene 425



(b) Gene 426



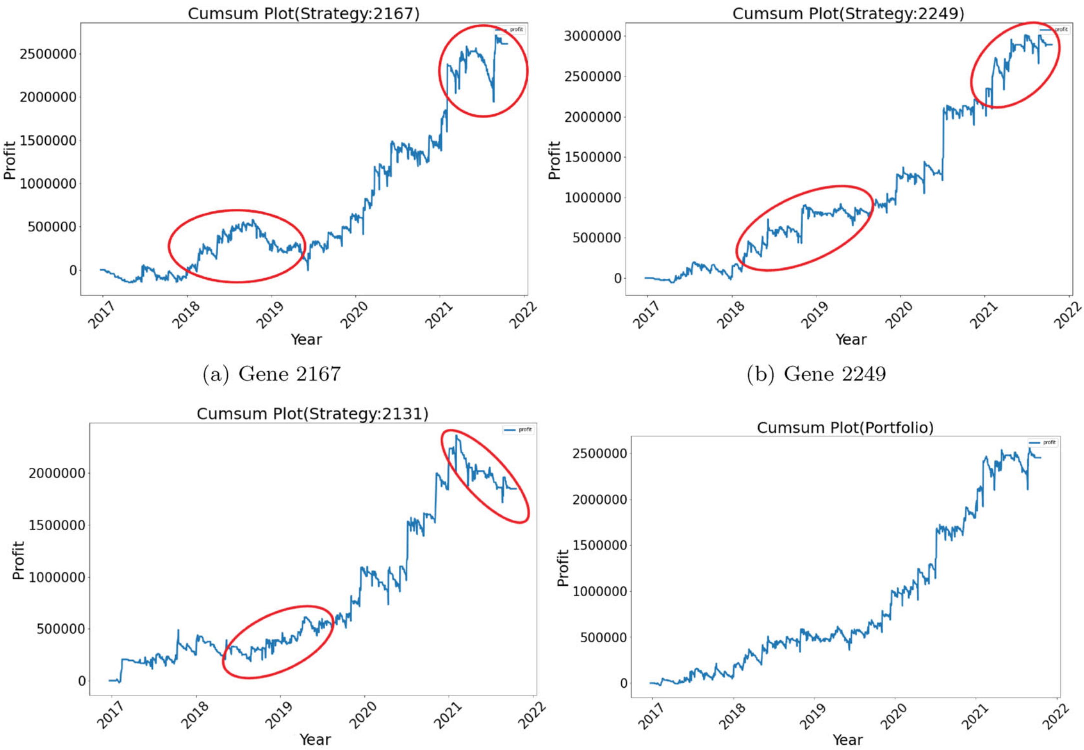
(c) The chromosome consists of gene 425 twice and gene 426.

Figure 9. The accumulated profit of MBS. (a) Gene 425. (b) Gene 426 and (c) The chromosome consists of gene 425 twice and gene 426.

Table 5. Performance of iterations for all strategies.

Number of iterations	Accumulated profit	Profit factor	Maximum drawdown	Win rate
1	1,909,770	1.25	−333,333	39.15
2	1,909,770	1.25	−333,333	39.15
3	1,909,770	1.25	−333,333	39.15
4	1,909,770	1.25	−333,333	39.15
5	2,660,800	1.3	−336,200	37.63
6	2,660,800	1.3	−336,200	37.63
7	2,660,800	1.3	−336,200	37.63
8	2,660,800	1.3	−336,200	37.63
9	1,672,200	1.19	−365,267	37.83
10	1,672,200	1.19	−365,267	37.83
11	2,401,070	1.28	−341,600	37.51
12	2,825,333	1.33	−329,749	39.02
13	2,825,333	1.33	−329,749	39.02
14	2,825,333	1.33	−329,749	39.02
15	2,825,333	1.33	−329,749	39.02
16	2,825,333	1.33	−329,749	39.02
17	3,046,000	1.33	−363,467	39.79

In complement to the visual analysis, various indicators such as the profit factor, accumulated profit, MDD and win rate throughout the PASS iteration are considered, as detailed in Table 5. These indicators provide supplementary insights into the performance and effectiveness of the strategies under evaluation.



(a) Gene 2167

(b) Gene 2249

(c) Gene 2131

(d) The chromosome consists of gene 2167, gene 2249 and gene 2131.

Figure 10. The accumulated profit of UMAS. (a) Gene 2167. (b) Gene 2249. (c) Gene 2131 and (d) The chromosome consists of gene 2167, gene 2249 and gene 2131.

As the number of iterations increases, an improvement is observed in performance. Notably, the seventeenth iteration stands out from the others, exhibiting the highest accumulated profit and win rate. However, it is worth mentioning that that iteration also demonstrates an increase in the MDD, highlighting a trade-off between profitability and drawdown mitigation.

For performance comparison, the primary performance metrics are as follows:

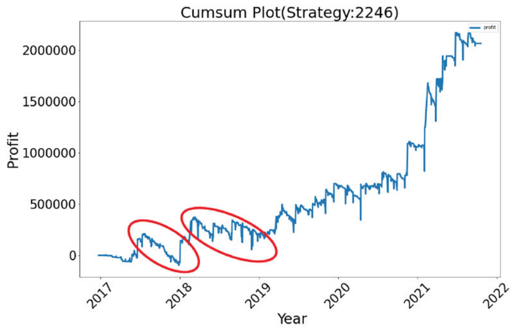
- Accumulated profit
- maxDDD

Additionally, other commonly used performance metrics are provided, including:

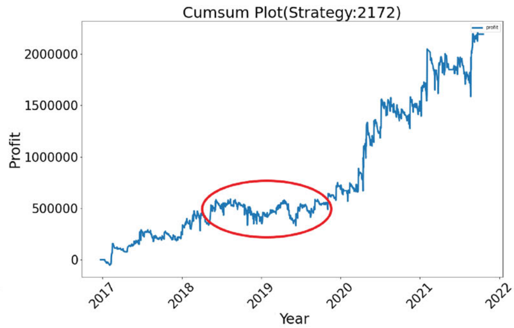
- Profit factor
- MDD
- Win rate

These metrics have been carefully selected to offer a thorough assessment of the algorithms' performance, addressing both profitability and risk. It is believed that including these metrics provides a well-rounded view of each algorithm's strengths and weaknesses, thereby meeting the criteria for a comprehensive experimental evaluation.

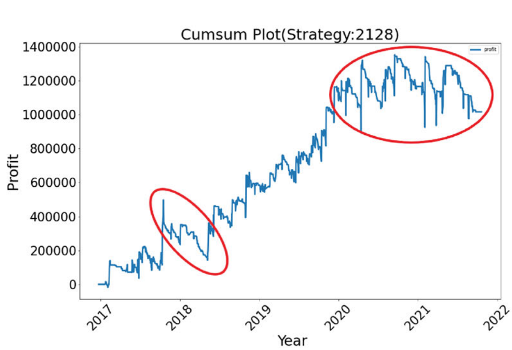
In Table 6, the performance of different algorithms is evaluated by assessing their ability to achieve the highest accumulated profit. It is evident that both the GA and differential evolution algorithms



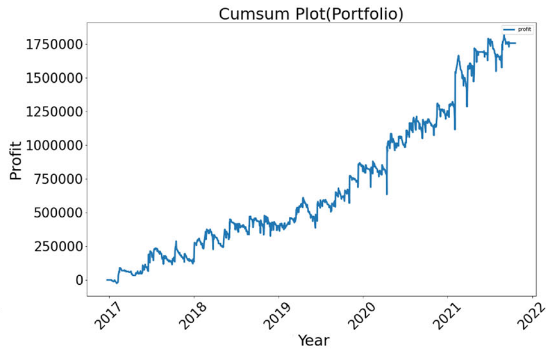
(a) Gene 2246



(b) Gene 2172

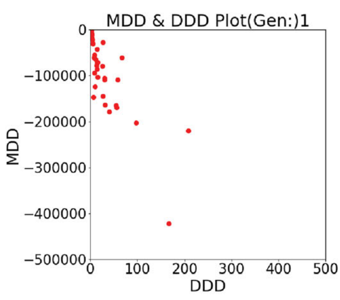


(c) Gene 2128

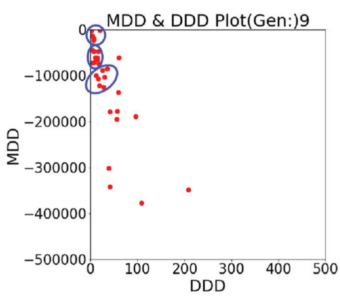


(d) The chromosome consists of gene 2246, gene 2172 and gene 2128.

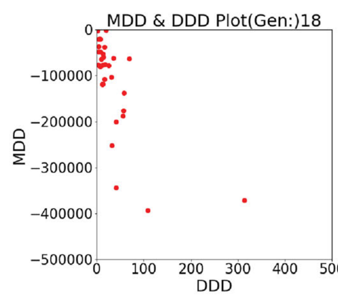
Figure 11. The accumulated profit of all strategies. (a) Gene 2246. (b) Gene 2172. (c) Gene 2128 and (d) The chromosome consists of gene 2246, gene 2172 and gene 2128.



(a) The first iteration



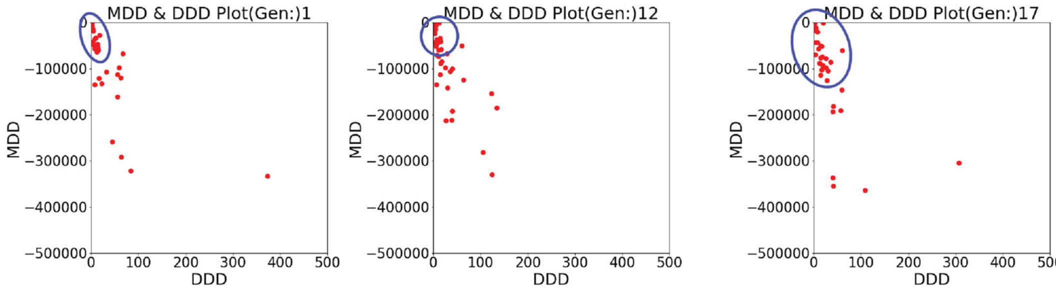
(b) The ninth iteration



(c) The eighteenth iteration

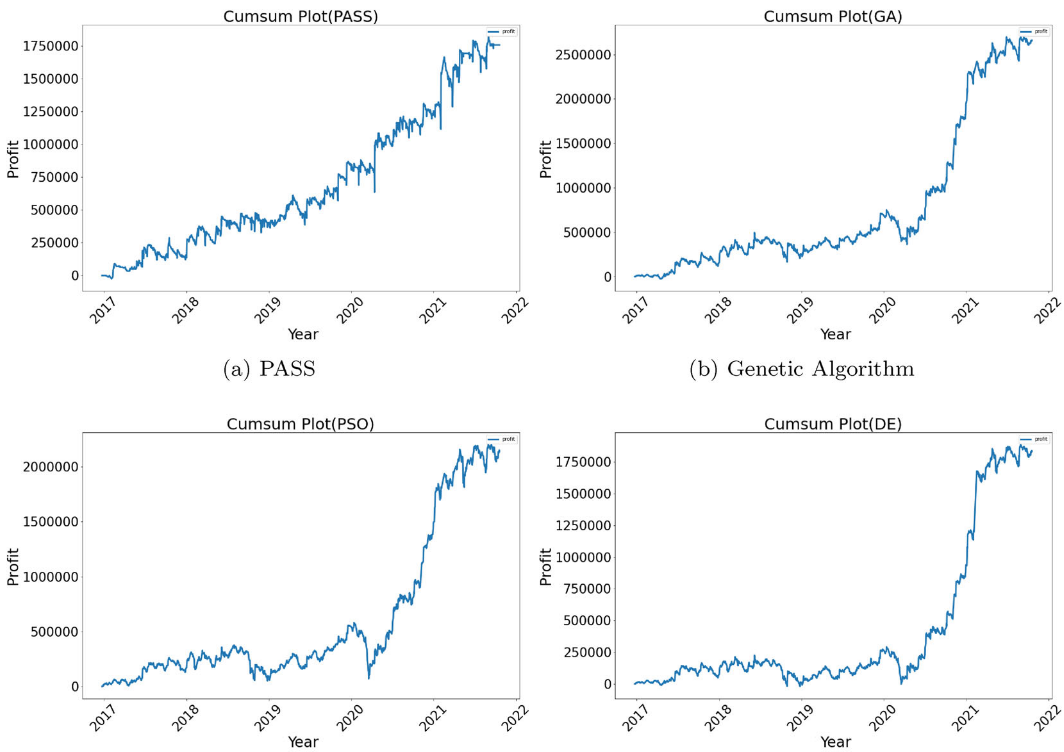
Figure 12. The iterations for ultimate moving average. (a) The first iteration. (b) The ninth iteration and (c) The eighteenth iteration.

excel in maximizing the profit factor. Among these algorithms, the GA demonstrates the highest profit. However, when comparing the PASS model with the differential evolution algorithm, a trade-off is observed. While the PASS model achieves a slightly lower accumulated profit (reduced by 4.1%), it significantly reduces the MDD by 73.9%. On the other hand, the particle swarm optimization algorithm exhibits the best win rate. Figure 14 shows four line charts of accumulated profit with each algorithm.



(a) The first iteration (b) The twelfth iteration (c) The seventeenth iteration

Figure 13. The iterations for all strategies. (a) The first iteration. (b) The twelfth iteration and (c) The seventeenth iteration.



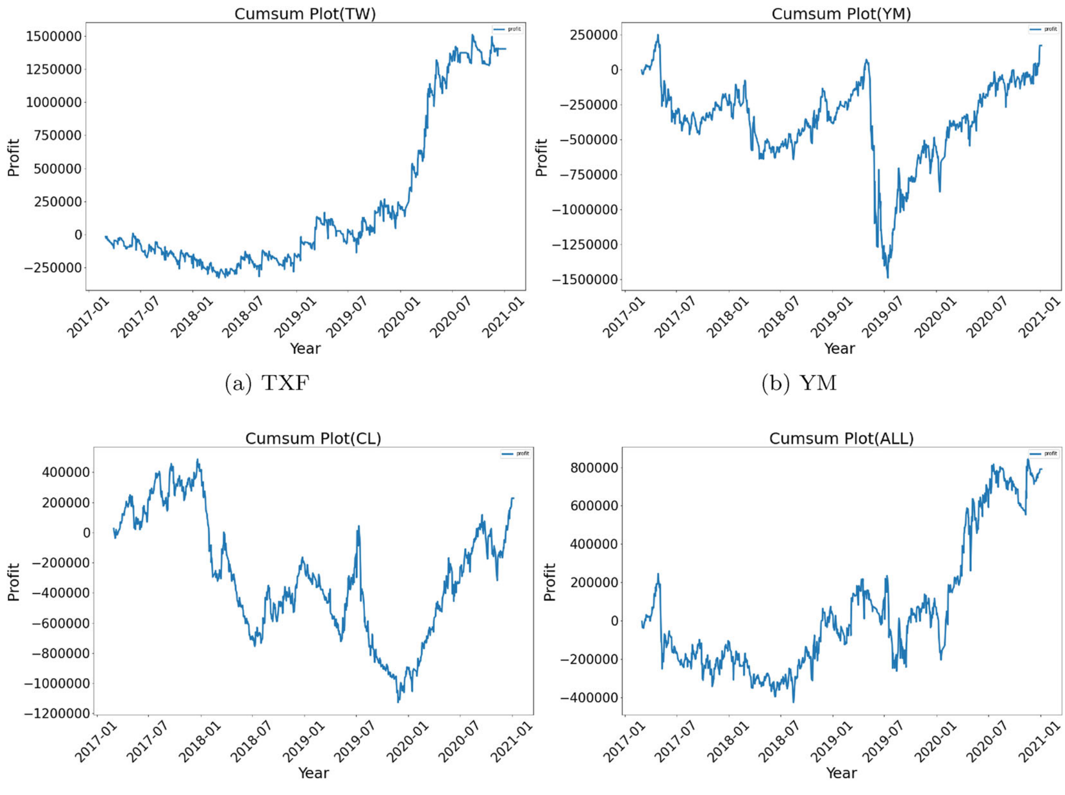
(a) PASS (b) Genetic Algorithm

(c) Particle Swarm Optimization (d) Differential Evolution

Figure 14. Four line charts of accumulated profit with each algorithm. (a) PASS. (b) Genetic algorithm. (c) Particle swarm optimization and (d) Differential evolution.

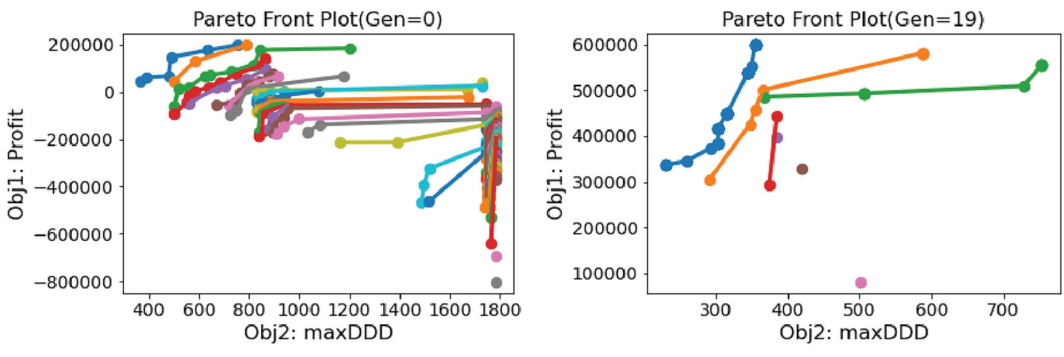
Table 6. Performance with PASS and efficient search algorithm.

Algorithm	Genes of the best chromosome	Accumulated profit	Profit factor	MDD	Win rate	maxDDD
PASS	[2246, 2172, 2128]	1,758,267	1.2	−379,933	36.52	98
GA	[2255, 2256, 420]	2,660,600	1.37	−387,467	40.45	315
PSO	[427, 421, 2256]	2,144,067	1.31	−510,733	50.98	300
DE	[2256, 425, 2280]	1,833,933	1.37	−293,800	43.5	376



(a) TXF (b) YM (c) WTI (d) TXF, YM, and WTI

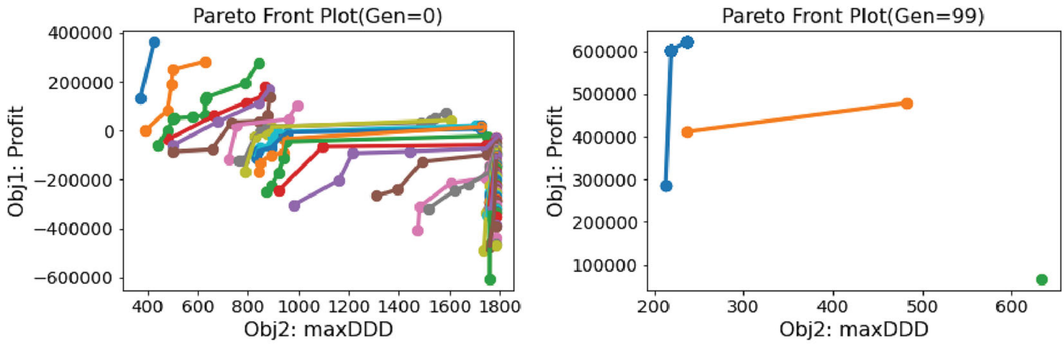
Figure 15. The moving window with the different futures markets. (a) TXF. (b) YM. (c) WTI and (d) TXF, YM and WTI.



(a) The first iteration (b) The eighteen iteration

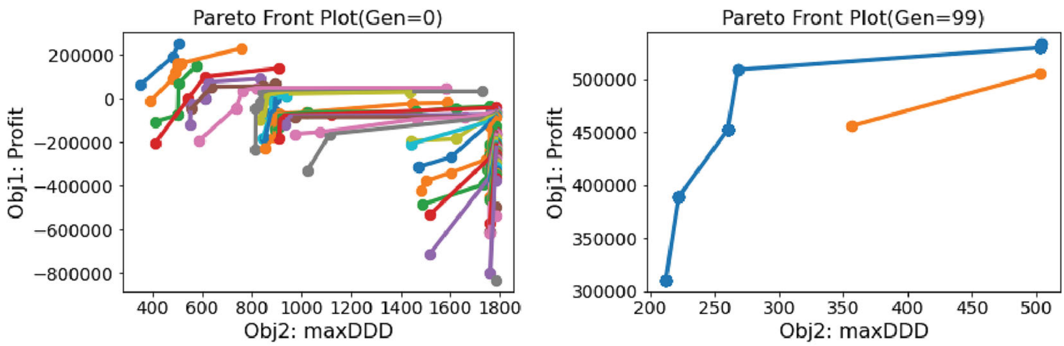
Figure 16. The number of genes is 3, the number of chromosomes is 100 and the number of iterations is 20. (a) The first iteration and (b) The eighteenth iteration.

An in-depth analysis of the daily profits of the PASS, GA, PSO and DE algorithms was conducted using the Friedman test. The results revealed a Friedman chi-squared value of 70.942 with three degrees of freedom and a p-value of 2.683e−15. These findings suggest a significant difference in the rankings of the algorithms, with the high chi-squared value indicating substantial variation between



(a) The first iteration (b) The ninety-eighth iteration

Figure 17. The number of genes is 3, the number of chromosomes is 100 and the number of iterations is 100. (a) The first iteration and (b) The ninety-eighth iteration.



(a) The first iteration (b) The ninety-eighth iteration

Figure 18. The number of genes is 10, the number of chromosomes is 100 and the number of iterations is 100. (a) The first iteration and (b) The ninety-eighth iteration.

Table 7. The post-hoc Nemenyi test for the PASS, GA, PSO and DE algorithms.

	PASS	GA	PSO
GA	0.000, 90	–	–
PSO	6.0e−14	0.000, 13	–
DE	9.6e−07	0.462, 16	0.028, 21

them, and the extremely low p-value providing strong evidence that these differences are statistically significant.

To explore these differences further, a post-hoc Nemenyi test was carried out, as summarized in Table 7. Based on the results, it can be concluded that PASS demonstrates a significant difference when compared to all other algorithms. Additionally, GA and PSO show a significant difference, as do PSO and DE. These results highlight that performance differences between the algorithms are notable in several cases, with PSO and PASS consistently differing from the others, while GA and DE appear to be more similar. The variation in performance is likely due to differences in how the algorithms handle chromosome selection.

Table 8. The performance of PASS applied in the futures market.

Futures	Genes of the best chromosome	Accumulated profit	Profit factor	MDD	Win rate	max-DDD
TXF	[2246, 2172, 2128]	1,758,267	1.2	−379,933	36.52	98
YM	[2309, 2514, 2519]	2,366,906	1.18	−798,400	51.29	190
WTI	[4409, 4614, 4619]	613,500	1.04	−1,835,900	52.79	653

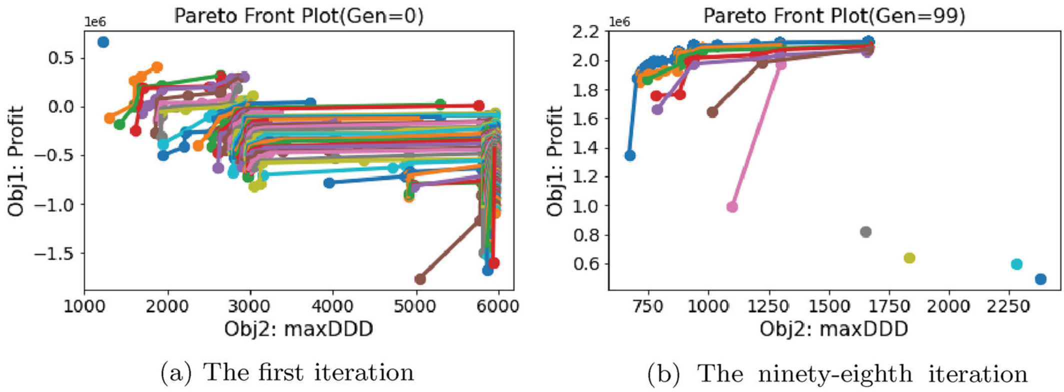


Figure 19. The number of genes is 10, the number of chromosomes is 500 and the number of iterations is 100. (a) The first iteration and (b) The ninety-eighth iteration.

To extend the application of the PASS model to other futures markets, the MBS and UMAS are employed via PASS. By utilizing historical data in Table 8, the optimal chromosome is identified. However, considering the high time correlation inherent in the futures market, the moving window method is adopted to ensure adaptability to market conditions. This approach involves partitioning the dataset into three-month windows for training and backtesting purposes, as shown in Figure 15. The best chromosome obtained from PASS is trained within each window and then applied to the subsequent window. The results across different futures markets validate the profitability of the chromosome, demonstrating the effectiveness of the moving window method.

In summary, the findings indicate that the TXF futures market proves to be the most favourable for implementing the UMAS via PASS. TXF demonstrates a reduction in drawdown from 2017 to 2018 and exhibits strong performance from 2019 to 2021. However, it is worth noting that both the UMAS and MBS exhibit relatively poor performance in YM and WTI owing to extended DDD and significant drawdown magnitudes, despite ultimately yielding profitable results. To achieve exceptional performance in YM and WTI, it is recommended to explore and develop more profitable quantitative trading strategies.

4.3. The hyperparameters of PASS

In this subsection, the objective is to determine the optimal number of genes, chromosomes and iterations for the PASS model. To facilitate the analysis, Pareto front charts were utilized.

By examining the charts from Figures 16 to 19, the elimination of poorly performing chromosomes is observed as the iterations progress. Conversely, superior chromosomes exhibit increased profitability and reduced DDD. Chromosomes at the same level on the Pareto front are considered to have comparable performance. As the number of iterations increases, the number of Pareto front solutions decreases and tends to cluster in the upper-left region of the charts. Moreover, it is found that increasing the number of genes in the chromosome leads to a slower convergence rate.

Based on the observations, the following three key conclusions are drawn.

- Increasing the number of genes in the chromosome is not advantageous for the PASS model.
- With an increased number of iterations, the clarity of the Pareto front levels improves.
- Excessive numbers of chromosomes and larger chromosome sizes result in a more complex Pareto front, making it challenging to determine the best-performing chromosome.

These insights provide valuable guidance for optimizing the configuration of genes, chromosomes and iterations in the PASS model.

5. Conclusions and discussion

In this study, the PASS model was introduced to construct portfolios of strategies that achieve sustained profit growth while effectively mitigating risks. To evaluate these strategy combinations, the DDD indicator was introduced, which measures the time taken to surpass previous profit peaks, highlighting the model's focus on consistent profitability. Crafting an optimal strategy combination posed challenges owing to the vast number of available strategies, the incorporation of TSPs, the moving window method and the extended backtesting period. To address complexity, the NSGA-II algorithm was employed to optimize the strategy selection process.

Strategy selection was based on two key features: profitability and maxDDD, ensuring that the selected strategy combination both generates profits and minimizes losses. Applying the PASS model, two strategies for the TXF market significantly reduced maxDDD during backtesting while maintaining consistent profits. A detailed comparison between the best investment strategy formed using the PASS method and three individual TSP strategies demonstrated notable improvements: a 13% reduction in DDD for the first TSP, a 19% reduction for the second TSP and an 11% reduction for the third TSP. Additionally, profitability increased by 7% for the first TSP, remained relatively unchanged at −0.15% for the second TSP, and increased by 11% for the third TSP.

However, achieving consistent profitability in the YM and WTI futures markets was challenging, primarily owing to the lack of profitable strategies within the TSP pool for these markets. To address the issue, it was recommended that the TSP pool be expanded with more profitable strategies, which would enable the identification of superior combinations capable of both amplifying profits and reducing drawdown using the PASS model.

In the experiments, the PASS model outperformed other search algorithms, such as GA, particle swarm optimization (PSO), and differential evolution (DE), particularly excelling in reducing maxDDD by 60%. Although the overall profit, profit factor and win rates did not show significant superiority, the model consistently demonstrated an upward trend in profitability. Additionally, optimal hyperparameters for applying the model to futures markets were identified, underscoring the importance of selecting a suitable TSP pool tailored to different markets, as strategies like UMAS and MBS may not be effective for the YM and WTI markets.

5.1. Limitations

The results of the study indicate that the PASS model is effective in constructing portfolios that balance profit maximization and risk minimization. The DDD indicator introduces a novel approach to evaluating the stability of trading strategies, offering deeper insights into long-term profitability and risk. This study contributes to the field of quantitative trading by providing a systematic approach to strategy selection and portfolio optimization. The use of MOEAs in this context allows for a more nuanced consideration of multiple objectives, moving beyond traditional single-objective optimization methods. The fact that the PASS model's capacity to reduce DDD and enhance profit stability has practical implications for traders and investors, fostering the development of more resilient strategies capable of withstanding market volatility.

However, this study has several limitations. One of the critical challenges in financial markets is their constant evolution, where past trends may not always reflect future movements. Over the past three years, financial markets have shown significant divergence, and in 2024, the Taiwan stock market experienced its most significant drop in history. This highlights the risk of relying solely on historical data for strategy development. The effectiveness of the PASS model is highly dependent on the quality and availability of historical market data. Incomplete or inaccurate data can adversely affect the model's performance. Additionally, the MOEA employed in the PASS model can be computationally intensive, particularly when applied to large datasets and extensive strategy pools, which results in long processing times and increased computational resource requirements.

Another limitation is the assumption that past market behaviour predicts future trends, which may not always hold true. Sudden market changes or black swan events can render the strategies ineffective. There is also a risk of overfitting, where the model excels in backtesting but struggles to generalize to new market conditions. The strategy pool in this study is also limited in scope, encompassing a restricted number of trading strategies. Expanding the pool to include a wider variety of strategies could potentially enhance the model's robustness and adaptability.

While GAs provide a powerful tool for portfolio optimization, they cannot guarantee profitability during structural market shifts. Therefore, it is essential to develop methods that detect when strategies within the PASS model remain effective and when they require adjustment or replacement to maintain profitability.

5.2. Future research directions

Addressing the limitations discussed in Section 5.1, future research should explore incorporating real-time data streams and adaptive algorithms to mitigate overfitting risks and improve the model's responsiveness to changing market conditions. Applying the PASS model to different asset classes, including equities, commodities and cryptocurrencies, could further validate the model and enhance its generalizability.

Another promising avenue for future research involves exploring methods to detect real-time market structure shifts. Integrating such detection mechanisms would allow for the identification of optimal strategy parameters, enabling better adaptation to evolving market conditions and improving long-term profitability. Developing tools for timely recognition of structural shifts is key to enhancing the robustness of portfolio strategies in dynamic financial markets.

In addition to addressing market structure shifts, future research should also consider the ethical implications of algorithmic trading. Although algorithmic trading presents opportunities for increased efficiency and profitability, it also raises concerns about fairness, transparency and potential market manipulation. Although the primary focus of this study has been on optimizing trading strategies using the PASS model, there is a need for future research to explore the ethical dimensions of these systems. Specifically, critical areas to investigate are the risks of unintended market manipulation, the impact of high-frequency trading on market volatility, and the importance of regulatory frameworks that promote responsible algorithmic trading practices.

Addressing these concerns requires the design of algorithmic trading systems that prioritize fairness and transparency. It includes developing mechanisms for monitoring trading activity to prevent dominance or manipulation by any single entity and ensuring that the decision-making processes of algorithms are transparent to both regulators and market participants. Future research should aim to incorporate safeguards within algorithmic models to prevent exploitation, ensuring that trading strategies comply with regulatory standards and contribute to fair market behaviour.

Abbreviations

DD	drawdown
DDD	drawdown duration

DE	differential evolution
maxDDD	maximum drawdown duration
MBS	momentum breakout strategy
MDD	maximum drawdown
MOEA	multi-objective evolutionary algorithm
NSGA-II	non-dominated sorting genetic algorithm-II
GA	genetic algorithm
PGAs	parallel genetic algorithms
PASS	portfolio analysis of selecting strategies
PSO	particle swarm optimization
TSP	trading strategy prototype
UMAS	ultimate moving average strategy
WFA	walk forward analysis

Nomenclature

Symbol	*Description*
β	A coefficient reflecting the acceptable level of loss
Δ	The windows of WFA
Amplitude	The difference between the highest price before the day and the lowest price before the day observed on the previous trading day
call	The call position price
Cdis	The sum of distances for all objectives
$Cum.P(t)$	The accumulated profit and loss at timestamp t
dis_M	The distance of the solution with M
Data	The original profit based on the timestamp
$H_{\text{timestamp}}$	The highest price in the timestamp
$L_{\text{timestamp}}$	The lowest price in the timestamp
MA	The average of the last k closest prices
momentum	Multiplying the amplitude with a coefficient
n	The number of chromosomes
N	The number of strategies included in the strategy pool
N_{tsp}	The number of strategies included in the TSP
pyramiding	The price of the subsequent long position
P_{i_j}	The profit of a specific sub-strategy in the ith TSP, TSP_i
$P(t)$	The unrealized/realized profit of the trading date at timestamp t
stop loss	Stop loss price
W_{ij}	From $Data_i$ to $Data_j$

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

Part of the funding for this research was provided by the National Science and Technology Council of Taiwan [grant no. NSTC 111-2221-E-027-115-MY3].

References

Aguirre, Alberto, Ricardo Medina, and Néstor Duque. 2020. “Machine Learning Applied in the Stock Market through the Moving Average Convergence Divergence (MACD) Indicator.” *Investment Management and Financial Innovations* 17 (4): 44–60. https://doi.org/10.21511/imfi.17(4).2020.05.

Bi, Wenhao, Jiuli Zhou, Junyi Shen, and An Zhang. 2024. "Optimization Method of Passive Omnidirectional Buoy Array in On-Call Anti-Submarine Search Based on Improved NSGA-II." *Ocean Engineering* 293:116655. https://doi.org/10.1016/j.oceaneng.2023.116655.

Bora, Teodoro Cardoso, Viviana Cocco Mariani, and Leandro dos Santos Coelho. 2019. "Multi-Objective Optimization of the Environmental-Economic Dispatch with Reinforcement Learning Based on Non-Dominated Sorting Genetic Algorithm." *Applied Thermal Engineering* 146:688–700. https://doi.org/10.1016/j.applthermaleng.2018.10.020.

Bradshaw, Noël-Ann, Chris Walshaw, Constantinos Ierotheou, and A. Kevin Parrott. 2009. "A Multi-Objective Evolutionary Algorithm for Portfolio Optimisation." In *Proceedings of the Adaptive and Emergent Behaviour and Complex Systems Convention*, 27–32. London: Society for the Study of Artificial Intelligence and Simulation of Behaviour. https://gala.gre.ac.uk/id/eprint/7104/.

Chan, Ernest P. 2009. *Quantitative Trading: How to Build Your Own Algorithmic Trading Business*. Hoboken, NJ: Wiley.

Chen, Chun-Hao, Cheng-Bon Lin, and Chao-Chun Chen. 2015. "Mining Group Stock Portfolio by Using Grouping Genetic Algorithms." In *Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC)*, 738–743. Piscataway, NJ: IEEE.

Choi, Jaehyung. 2021. "Maximum Drawdown, Recovery, and Momentum." *Journal of Risk and Financial Management* 14 (11): 542. https://doi.org/10.3390/jrfm14110542.

Chou, Yao-Hsin, Yu-Chi Jiang, Yi-Rui Hsu, Shu-Yu Kuo, and Sy-Yen Kuo. 2021. "A Weighted Portfolio Optimization Model Based on the Trend Ratio, Emotion Index, and ANGQTS." *IEEE Transactions on Emerging Topics in Computational Intelligence* 6 (4): 867–882. https://doi.org/10.1109/TETCI.2021.3118041.

Chou, Yao-Hsin, Yu-Chi Jiang, and Shu-Yu Kuo. 2021. "Portfolio Optimization in Both Long and Short Selling Trading Using Trend Ratios and Quantum-Inspired Evolutionary Algorithms." *IEEE Access*9:152115–152130. https://doi.org/10.1109/ACCESS.2021.3126652.

Colaço, Andrey Barbosa, Viviana Cocco Mariani, Mohamed Reda Salem, and Leandro dos Santos Coelho. 2022. "Maximizing the Thermal Performance Index Applying Evolutionary Multi-Objective Optimization Approaches for Double Pipe Heat Exchanger." *Applied Thermal Engineering* 211:118504. https://doi.org/10.1016/j.applthermaleng.2022.118504.

Daskalakis, George, and Raphael N. Markellos. 2008. "Are the European Carbon Markets Efficient?" *Review of Futures Markets* 17 (2): 103–128.

Deb, Kalyanmoy, Amrit Pratap, Sameer Agarwal, and T. A. M. T. Meyarivan. 2002. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation* 6 (2): 182–197. https://doi.org/10.1109/4235.996017.

De Jong, Kenneth A., and William M. Spears. 1989. "Using Genetic Algorithms to Solve NP-Complete Problems." In *Proceedings of the Third International Conference on Genetic Algorithms*, 124–132. San Francisco, CA: Morgan Kaufmann. https://www.researchgate.net/publication/201976504_Using_Genetic_Algorithms_to_Solve_NP-Complete_Problems.

Dorigo, M., V. Maniezzo, and A. Colorni. 1996. "Ant System: Optimization by a Colony of Cooperating Agents." *IEEE Transactions on Systems Man and Cybernetics Part B—Cybernetics* 26 (1): 29–41. https://doi.org/10.1109/TSMCB.3477.

Drenovak, Mikica, Vladimir Ranković, Branko Urošević, and Ranko Jelic. 2022. "Mean-Maximum Drawdown Optimization of Buy-and-Hold Portfolios Using a Multi-Objective Evolutionary Algorithm." *Finance Research Letters* 46:102328. https://doi.org/10.1016/j.frl.2021.102328.

Fang, Fan, Carmine Ventre, Michail Basios, Leslie Kanthan, David Martinez-Rego, Fan Wu, and Lingbo Li. 2022. "Cryptocurrency Trading: A Comprehensive Survey." *Financial Innovation* 8 (1): 13. https://doi.org/10.1186/s40854-021-00321-6.

Fernandez, Eduardo, Jorge Navarro, Efrain Solares, and Carlos Coello Coello. 2019. "A Novel Approach to Select the Best Portfolio Considering the Preferences of the Decision Maker." *Swarm and Evolutionary Computation* 46:140–153. https://doi.org/10.1016/j.swevo.2019.02.002.

Galuzio, Paulo Paneque, Emerson Hochsteiner de Vasconcelos Segundo, Leandro dos Santos Coelho, and Viviana Cocco Mariani. 2020. "MOBOpt—Multi-Objective Bayesian Optimization." *SoftwareX*12:100520. https://doi.org/10.1016/j.softx.2020.100520.

Glasserman, Paul, Philip Heidelberger, and Perwez Shahabuddin. 2002. "Portfolio Value-at-Risk with Heavy-Tailed Risk Factors." *Mathematical Finance* 12 (3): 239–269. https://doi.org/10.1111/mafi.2002.12.issue-3.

Gunantara, Nyoman. 2018. "A Review of Multi-Objective Optimization: Methods and Its Applications." *Cogent Engineering* 5 (1): 1502242. https://doi.org/10.1080/23311916.2018.1502242.

Hadka, David, and Patrick Reed. 2012. "Diagnostic Assessment of Search Controls and Failure Modes in Many-Objective Evolutionary Optimization." *Evolutionary Computation* 20 (3): 423–452. https://doi.org/10.1162/EVCO_a_00053.

Holland, J. H. 1992. "Genetic Algorithms." *Scientific American* 267 (1): 66–73. https://www.jstor.org/stable/24939139.

Huizinga, Joost, and Jeff Clune. 2022. "Evolving Multimodal Robot Behavior Via Many Stepping Stones with the Combinatorial Multiobjective Evolutionary Algorithm." *Evolutionary Computation* 30 (2): 131–164. https://doi.org/10.1162/evco_a_00301.

Israelsen, Craig. 2005. “A Refinement to the Sharpe Ratio and Information Ratio.” *Journal of Asset Management* 5:423–427. https://doi.org/10.1057/palgrave.jam.2240158.
Karaboga, Dervis. 2005. “An Idea Based on Honey Bee Swarm for Numerical Optimization.” *Technical Report-tr06*. Kayseri, Turkey: Computer Engineering Department, Engineering Faculty, Erciyes University. https://abc.erciyes.edu.tr/pub/tr06_2005.pdf.
Kennedy, James, and Russell Eberhart. 1995. “Particle Swarm Optimization.” In *Proceedings of the International Conference on Neural Networks (ICNN’95)*, Vol. 4, 1942–1948. Piscataway, NJ: IEEE. https://doi.org/10.1109/ICNN.1995.488968.
Kleeman, Mark P., Benjamin A. Seibert, Gary B. Lamont, Kenneth M. Hopkinson, and Scott R. Graham. 2012. “Solving Multicommodity Capacitated Network Design Problems Using Multiobjective Evolutionary Algorithms.” *IEEE Transactions on Evolutionary Computation* 16 (4): 449–471. https://doi.org/10.1109/TEVC.2011.2125968.
Li, Shimin, Huiling Chen, Mingjing Wang, Ali Asghar Heidari, and Seyedali Mirjalili. 2020. “Slime Mould Algorithm: A New Method for Stochastic Optimization.” *Future Generation Computer Systems*111:300–323. https://doi.org/10.1016/j.future.2020.03.055.
Lin, Ping-Chen, and Po-Chang Ko. 2009. “Portfolio Value-at-Risk Forecasting with GA-Based Extreme Value Theory.” *Expert Systems with Applications* 36 (2): 2503–2512. https://doi.org/10.1016/j.eswa.2008.01.086.
Ma, Dongfang, Siyuan Zhou, Yueyi Han, Weihao Ma, and Hongxun Huang. 2024. “Multi-Objective Ship Weather Routing Method Based on the Improved NSGA-III Algorithm.” *Journal of Industrial Information Integration* 38:100570. https://doi.org/10.1016/j.jii.2024.100570.
Magdon-Ismail, M., A. F. Atiya, A. Pratap, and Y. S. Abu-Mostafa. 2004. “On the Maximum Drawdown of a Brownian Motion.” *Journal of Applied Probability* 41 (1): 147–161. https://doi.org/10.1239/jap/1077134674.
Markowitz, Harry. 1952. “Portfolio Selection.” *Journal of Finance* 7 (1): 77.
Meghwani, Suraj S., and Manoj Thakur. 2018. “Multi-Objective Heuristic Algorithms for Practical Portfolio Optimization and Rebalancing with Transaction Cost.” *Applied Soft Computing* 67:865–894. https://doi.org/10.1016/j.asoc.2017.09.025.
Moosavi, S. H. S., and V. K. Bardsiri. 2019. “Poor and Rich Optimization Algorithm: A New Human-Based and Multi Populations Algorithm.” *Engineering Applications of Artificial Intelligence* 86:165–181. https://doi.org/10.1016/j.engappai.2019.08.025.
Nabipour, Mojtaba, Pooyan Nayyeri, Hamed Jabani, S. Shahab, and Amir Mosavi. 2020. “Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis.” *IEEE Access* 8: 150199–150212. https://doi.org/10.1109/ACCESS.2020.3015966.
Nayak, Sarat Chandra, and Bijan Bihari Misra. 2018. “Estimating Stock Closing Indices Using a GA-Weighted Condensed Polynomial Neural Network.” *Financial Innovation* 4 (1): 21. https://doi.org/10.1186/s40854-018-0104-2.
Pardo, Robert, ed. 2012. *The Evaluation and Optimization of Trading Strategies*. Chichester, UK: Wiley. https://doi.org/10.1002/9781119196969.
Qian, Yihe, and Jinpeng Wang. 2024. “Multi-Period Portfolio Optimization: A Parallel NSGA-III Algorithm with Real-World Constraint.” *Finance Research Letters*60:104868. https://doi.org/10.1016/j.frl.2023.104868.
Sebastião, Helder, and Pedro Godinho. 2021. “Forecasting and Trading Cryptocurrencies with Machine Learning under Changing Market Conditions.” *Financial Innovation* 7 (1): 3. https://doi.org/10.1186/s40854-020-00217-x.
Shabani, Amir, Behrouz Asgarian, Miguel Salido, and Saeed Asil Gharebaghi. 2020. “Search and Rescue Optimization Algorithm: A New Optimization Method for Solving Constrained Engineering Optimization Problems.” *Expert Systems with Applications* 161:113698. https://doi.org/10.1016/j.eswa.2020.113698.
Srinivas, N., and Kalyanmoy Deb. 1994. “Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms.” *Evolutionary Computation* 2 (3): 221–248. https://doi.org/10.1162/evco.1994.2.3.221.
Sulaiman, Mohd Herwan, Zuriani Mustaffa, Mohd Mawardi Saari, and Hamdan Daniyal. 2020. “Barnacles Mating Optimizer: A New Bio-Inspired Algorithm for Solving Engineering Optimization Problems.” *Engineering Applications of Artificial Intelligence* 87:103330. https://doi.org/10.1016/j.engappai.2019.103330.
Tsang, Edward, and Jun Chen. 2018. “Regime Change Detection Using Directional Change Indicators in the Foreign Exchange Market to Chart Brexit.” *IEEE Transactions on Emerging Topics in Computational Intelligence* 2 (3): 185–193. https://doi.org/10.1109/TETCI.2017.2775235.
Vijh, Mehar, Deeksha Chandola, Vinay Anand Tikkiwal, and Arun Kumar. 2020. “Stock Closing Price Prediction Using Machine Learning Techniques.” *Procedia Computer Science* 167:599–606. https://doi.org/10.1016/j.procs.2020.03.326.
Wu, Mu-En, Jia-Hao Syu, and Chien-Ming Chen. 2022. “Kelly-Based Options Trading Strategies on Settlement Date Via Supervised Learning Algorithms.” *Computational Economics* 59 (4): 1627–1644. https://doi.org/10.1007/s10614-021-10226-2.
Xiong, Zhuoran, Xiao-Yang Liu, Shan Zhong, Hongyang Yang, and Anwar Walid. 2018. “Practical Deep Reinforcement Learning Approach for Stock Trading,” 1–7. arXiv: 1811.07522.
Zhao, Weiguo, Liying Wang, and Zhenxing Zhang. 2019. “Supply–Demand-Based Optimization: A Novel Economics-Inspired Algorithm for Global Optimization.” *IEEE Access* 7: 73182–73206. https://doi.org/10.1109/ACCESS.2019.2918753.