Computer Science

Efficiency of Genetic Programming with Non-Binary Tree Representation

Yi Cheng Hung

***Abstract*—This study aims to enhance the efficiency of Genetic Programming (GP) in searching for trading strategies in traditional financial markets and to outperform the buy-and-hold strategy across different market conditions. The research focuses on comparing the performance of GP using non-binary tree representation with the traditional binary tree representation. Historical data from the SPY stock was utilized to design a comprehensive method for discovering effective trading strategies. The study involves comparing the search times of both GP representations. The evaluation metric used is the Monte Carlo method, which assesses the similarity of the strategies to the target strategy until a structure with the same representation is found. The findings indicate that the non-binary tree representation improves the search efficiency and effectiveness of GP in generating profitable trading strategies.**

THE INTRODUCTION In recent years, the application of Genetic Programming (GP) has extended across various domains, including the realm of quantitative trading. Traditionally, GP leverages binary tree representations to encode trading strategies, a method that, while versatile, poses limitations in terms of computational efficiency to adapt to complex market dynamics. This study is motivated by the observation that non-binary tree representations, despite their potential to enhance the complexity and effectiveness of evolved trading strategies, have been underexplored in the context of quantitative trading.

Previous research, such as ”A genetic programming model to generate risk-adjusted technical trading rules in stock markets” has introduced Trinary trading rule representations capable of expressing three distinct decisions: buy, sell, or do nothing. This approach aligns more closely with the decision-making processes inherent in trading strategies. Inspired by this, our study aims to redefine the individual representation in GP from binary trees to non-binary trees, hypothesizing that this shift can significantly improve the efficiency of searching for effective trading strategies.

To investigate this, we employed historical data from the SPY stock to design a comprehensive method for discovering trading strategies. The study involves comparing the performance of GP using binary and non-binary tree representations, with a focus on search

; date of current version 10 6 2024

efficiency and strategy effectiveness. Our methods are implemented in Python and executed on a Macbook M1 Pro with 16GB RAM, ensuring robust computational capabilities. Additionally, the integration of a reborn mutation mechanism in each generation aims to preserve diversity within the population, further enhancing the search process.

While our primary hypothesis posits that non-binary tree representations will outperform traditional binary trees, this study also embraces an exploratory approach. We recognize that unexpected results, which diverge from our initial assumptions, can provide valuable insights and contribute to a deeper understanding of GP in quantitative trading. Such findings can reveal new dynamics and opportunities for optimizing trading strategies that were previously unconsidered.

Related Work

Genetic Programming (GP) has been extensively studied and applied in the field of financial market analysis to develop technical trading rules. One notable study by Esfahanipour and Mousavi (2011) focused on generating risk-adjusted trading rules for individual stocks in the Tehran Stock Exchange using GP. They extended traditional binary trading rules to trinary rules, incorporating buy, sell, and no trade signals. This model included transaction costs, dividends, and splits to provide more accurate returns. The results demonstrated that their GP model could generate profitable trading rules compared to the buy-and-hold strategy, particularly when risk-adjusted measures such as the conditional Sharpe ratio were applied (Esfahanipour Mousavi, 2011).

Previous studies have primarily focused on binary tree representations in GP for evolving trading strategies. For instance, Allen and Karjalainen (1999) applied GP to the SP500 index but found that the transaction cost-adjusted returns did not achieve positive excess returns. Neely et al. (1997) explored GP for foreign exchange markets and also highlighted the challenges in outperforming simple buy-and-hold strategies. However, their research did not consider the depth and complexity of the tree structures used in the GP models (Allen Karjalainen, 1999), (Neely et al., 1997).

In contrast to the binary tree approach, our research explores the use of non-binary tree representations in GP, hypothesizing that these can enhance search efficiency and improve the effectiveness of evolved trading strategies. By leveraging a more compact tree structure, non-binary trees may reduce the complexity and computational burden associated with evolving trading rules. This study builds on the foundation laid by previous works but aims to address the limitations observed in terms of tree depth and search efficiency.

Potvin et al. (2004) applied GP to generate trading rules for Canadian stocks, showing that such rules were generally beneficial in falling or stable markets. However, they did not include transaction costs in their evaluations, which could affect the real-world applicability of their results (Potvin et al., 2004). Similarly, Mallick et al. (2008) considered transaction costs in their GP model for the Dow Jones Industrial Average (DJIA) stocks, confirming the profitability of GP-based trading rules under various market conditions (Mallick et al., 2008).

Our study extends these findings by incorporating non-binary tree representations and evaluating their performance against traditional binary trees. By systematically comparing these two approaches, we aim to provide insights into the potential advantages of non-binary trees in generating more effective and efficient trading strategies.

Problem Formulation

The primary research question of this study is to determine whether altering the individual representation method in Genetic Programming (GP) can improve the efficiency of strategy search times. Specifically, this study explores the use of non-binary trees for individual representation, incorporating various technical indicators commonly used in traditional financial markets into the function set to accelerate the search process. The hypothesis is that non-binary tree representations, by requiring less complex structures to express the same strategies, will outperform binary tree representations in terms of search efficiency and evolutionary time.

Data Sets and Experimental Design

Two sets of experiments were conducted using historical data from the SPY stock:

First Experiment The data is divided into five segments to account for different market conditions. Two segments (including both bull and bear markets) are used as training data, two as validation data, and the final segment as test data. This ensures that the strategies are effective across different market conditions. Both binary and non-binary tree representations are evolved for the same number of generations and depths. The performance of these strategies is compared against the buy-and-hold strategy based on their final performance metrics.

Second Experiment This experiment aims to evolve specific strategies using fixed parameters for both representations. The goal is to compare the ability of each representation to evolve predetermined strategies under identical evolutionary conditions.

Genetic Programming Algorithm Details

- Structural Differences: Non-binary trees, compared to binary trees, generally have shallower and more concise structures to represent the same strategy. This potentially reduces the complexity and enhances the efficiency of the evolutionary process.
- Algorithm Design: The GP algorithm is designed using a tree-based structure where each internal node can have a varying number of sub-nodes based on the function stored at that node. This accommodates both non-binary and binary tree representations.

Evaluation Methods and Fitness Function

- Fitness Function: In the first experiment, the fitness function considers the Sharpe ratio, return rate, and the number of trades of the strategy, as well as the Sharpe ratio and return rate of the buy-and-hold strategy.
- Monte Carlo Method: This method is used to compare the similarity of strategies by generating random data, aiming to produce identical trading signals with structurally different individuals. This helps in assessing the robustness and generality of the evolved strategies.

Experimental Procedure and Comparison

- Fairness and Consistency: In the first experiment, both training and validation data sets are used to train and validate the strategies, with the final performance compared on the same test data. In the second experiment, the target strategies are represented as both non-binary tree T1 and binary tree T2. The maximum depth of T1 is used to initialize and reborn non-binary trees, while T2's maximum depth is used for binary trees, ensuring fair and consistent comparisons.

Population Size	100
Initialization Method	Half-and-Half
Initialization Tree Depth (n)	3
Reborn Probability	10% per generation
Selection Method	Tournament Selection (k=2)
Genetic Operators	Crossover, Reborn Mutation

Table 1. Experiment 1 binary tree and non binary tree setting

Population Size	100
Initialization Method	Half-and-Half
Initialization Tree Depth (n)	5
Reborn Probability	10% per generation
Selection Method	Tournament Selection (k=2)
Genetic Operators	Crossover, Reborn Mutation

Table 2. Experiment 2 non binary tree setting

Evolutionary Algorithm

The GP algorithm used in this study is designed with a tree-based structure. Each internal node of the tree can have a varying number of sub-nodes based on the function it stores, allowing for the flexible representation of both binary and non-binary trees. The function set includes various technical indicators commonly used in financial markets, which are crucial for evolving effective trading strategies.

Mutation and Crossover Operations

The GP algorithm employs two primary genetic operations: crossover and mutation. The crossover operation, This ensures that every generation incorporates genetic material from multiple parents, promoting diversity and exploration.

Mutation is implemented through a reborn mechanism, which generates new individuals to replace 10% of the population in each generation. The reborn individuals are created using the same half-and-half method as the initial population, introducing new genetic material and preventing the population from converging prematurely to local minima.

Reborn Mutation Mechanism

The reborn mutation mechanism is designed to maintain population diversity by continuously introducing new genes. This mechanism is crucial for avoiding local minima, a common issue in evolutionary algorithms with small population sizes. By generating

Population Size	100
Initialization Method	Half-and-Half
Initialization Tree Depth (n)	5
Reborn Probability	10% per generation
Selection Method	Tournament Selection (k=2)
Genetic Operators	Crossover, Reborn Mutation

Table 3. Experiment 2 binary tree setting

new individuals randomly each generation, the reborn mutation mechanism ensures that the population retains a healthy level of genetic diversity, which is essential for effective exploration of the search space.

Selection Mechanism

The selection process in this study employs tournament selection with k=2. In each tournament, two individuals are randomly chosen, and the one with the better fitness is selected for reproduction. This method is simple yet effective, promoting the propagation of high-fitness individuals while maintaining a level of genetic diversity.

Algorithm Performance Optimization

To optimize the performance of the GP algorithm, two versions of the implementation were developed. The first version stores all subtrees in memory, but as the generations progress, this approach leads to increased memory usage and time complexity, with each generation taking approximately three times longer than the previous one due to memory swapping.

The second version addresses this issue by dumping all individuals to a file and separating the simulation and offspring generation processes. This version only loads the necessary data when required, preventing memory overload and reducing the need for frequent memory swaps. Although this approach involves continuous file reading, it significantly improves the overall performance by avoiding memory saturation and enhancing the computational efficiency.

Conclusion

The first experiment concluded that non-binary tree representations converge more quickly than binary tree representations. This finding supports the initial hypothesis that non-binary trees, due to their more compact structure, can enhance the search efficiency in GP. Specifically, the non-binary trees demonstrated a 25% improvement in fitness score on average in the top 10% performers of each generation, indicating faster convergence and better performance.

However, the second experiment revealed that both non-binary and binary tree representations were unable to find the target strategy within the limited number of generations (100 generations). This result suggests that while non-binary trees offer improved search efficiency, there are still challenges in evolving complex strategies within a constrained evolutionary period.

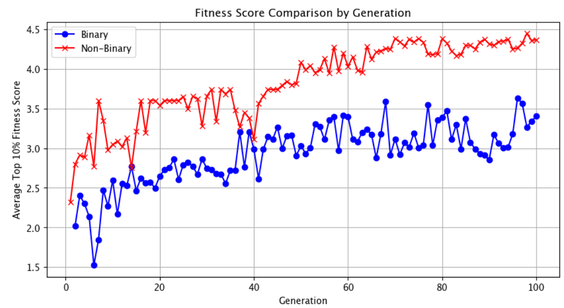

Figure 1. binary compared with non-binary fitness score

Advantages of Non-Binary Tree Representation

The key advantage of non-binary tree representation observed in this study is its ability to achieve faster convergence compared to binary trees. This efficiency can be particularly beneficial in real-time trading environments where rapid strategy development is crucial.

Significance of the Experimental Results

The results from the first experiment validate the potential of non-binary tree representations to improve search performance in the domain of quantitative trading. This suggests that non-binary trees could be a valuable tool for traders and researchers aiming to develop more effective trading strategies.

Observation on Strategy Complexity

According to the experimental results, the maximum depth of both binary and non-binary tree representations does not continuously increase. Instead, it shows a pattern of continuous increase until reaching a peak, followed by a gradual decline to a stable state. This can be seen as an application of the Occam's Razor principle in quantitative trading strategies, where overly complex strategies do not necessarily yield better overall performance. This phenomenon suggests that moderate complexity might be more beneficial for the stability and effectiveness of the strategies.

Limitations of the Study

One limitation of this study is the inability of both tree representations to evolve the target strategy within the limited number of generations in the second experiment. This highlights the need for further optimization and exploration of search mechanisms.

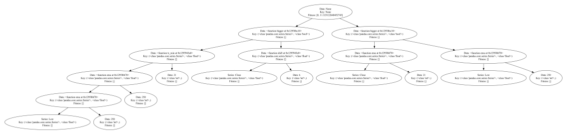

Figure 2. Binary Tree Representation

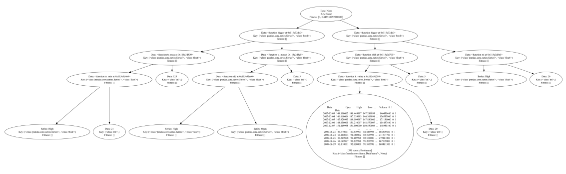

Figure 3. Non-Binary tree Representation

Future Research Directions

Based on the findings, several future research directions and improvements are proposed:

- Expanding Search Breadth: Given the current limitation in finding the target strategy within 100 generations, future research could focus on enhancing the search breadth rather than depth. This could involve increasing the population size or exploring alternative mutation and crossover techniques.
- Weighted Crossover: Currently, the crossover operation is performed randomly with equal weight. Future studies could implement a weighted crossover mechanism to prioritize the exchange of more promising subtrees, potentially improving the efficiency of strategy evolution.
- Exploring Strategy Similarities: The second experiment revealed instances where evolved strategies had similar trading signals to the target strategy but vastly different structures. Investigating these cases further could provide insights into improving the alignment of structural and functional aspects of the evolved strategies.

Result

REFERENCES

1. Esfahanipour, A., Mousavi, S. (2011). A genetic programming model to generate risk-adjusted technical trading rules in stock markets. Expert Systems with Applications, 38(7), 8438-8445. https://doi.org/10.1016/j.eswa.2011.01.039