Optimizing Stock Market Trading Strategies through Evolutionary Computation

Timothy D. Hall
University of Georgia
tdh22457@uga.edu

Abstract— **The inherent complexity of financial markets presents a pressing need for innovative approaches for predicting its outcomes. Traditional models often fall short in capturing the intricate and ever-changing patterns that dictate market movements. Additionally, many models focus on predicting where the stock will go on a long-term basis, while neglecting the short-term patterns, presenting the need for a prediction model that can manage the intricacies of short-term price movements in the stock market. The goal of this research is to create and test multiple evolutionary algorithms whose objective is to track small, transitory price movements in the stock market effectively and consistently. Experimental results demonstrate that using both Genetic Algorithms (GA) and Evolutionary Programming (EP) Algorithms result in the development of optimal trading strategies that are not only profitable in a controlled setting, but also outperform traditional algorithmic trading methods.**

Index Terms— **Crossover Operators, Evolutionary Computation, Evolutionary Programming, Genetic Algorithms, Genetic Operators, Mutation Operators**

I. INTRODUCTION

THE stock market is a dynamic and intricate landscape that is influenced by a multitude of factors such as market trends, economic indicators, and geopolitical events. Financial analysts, researchers, and traders devote extraordinary amounts of time trying to navigate the complex terrain of stock exchanges to make informed predictions for future trading. Traditional strategies using algorithmic trading methods often fall short in capturing the nuanced patterns characteristic of market dynamics. This gap creates the need for Artificial Intelligence techniques to develop adaptive strategies that can navigate this multifaceted environment.

Intraday trading, with its myriad of parameters to consider and assign value to for arriving at an outcome, epitomizes one of the most unpredictable trading patterns. This concept is not unlike the complexity of ecological systems in which an individual must integrate multiple variables within their environment to make decisions that better the odds of their survival. Evolutionary computation exploits this model by mimicking the process of natural selection, characterized by the concept that the most fit individuals will survive and propagate, and uses this process to enhance trading strategies in the competitive stock market environment.

In the world of stock market prediction and algorithmic trading, there is no shortage of AI and machine learning methodologies, many of them utilizing evolutionary methods. There are many popular programs that range from traditional models to more sophisticated deep learning architectures such as Recurrent Neural Networks (RNNs) and convolutional neural networks (CNNs). Techniques like these often use a combination of multiple frameworks integrated into one approach, leaving evolutionary computational methodologies to play a supporting role. These methodologies, including Genetic Algorithms and Evolutionary Programming, are usually used for aiding in parameter optimization, feature selection, or helping to structure neural networks. The research conducted in this paper is distinct in its emphasis on employing Genetic Algorithms and Evolutionary Programming as the foundational concepts for the trading algorithm. This unique approach aims to leverage the inherent adaptive and explorative capabilities of Genetic Algorithms and Evolutionary Programming as the sole framework to predict intraday trading patterns, rather than relying on them as supplements within a more extensive AI-driven trading system.

Through empirical analyses, back testing, and forward testing, this paper demonstrates the efficacy of evolutionary computation algorithms to optimize stock market trading strategies. The outcomes of this research offer insights into the practical application of evolutionary computation techniques within the domain of financial markets and demonstrate their robustness and adaptability in volatile market conditions.

Timothy D. Hall is with the University of Georgia, School of Computing, Athens, GA 30602 USA (e-mail: tdh22457@uga.edu).

II. RELATED WORKS

Genetic Algorithms and Evolutionary Programming have captivated considerable interest for optimizing trading strategies within financial markets, lauded for their adeptness in navigating expansive solution spaces. A pioneering illustration of their application in market prediction emerged through a study conducted by Badawy et al. [1]. This study exemplified the effective integration of GA methodologies, encompassing mutation and crossover events, to iteratively evolve trading strategies across generations. Remarkably, this approach yielded a convergence towards profitable trading strategies within the domain of the Egyptian Stock Market.

However, there has been a marked shift in the application of GA for the use in financial markets. Recent models tend to utilize the evolutionary techniques seen in both GA and EP to optimize only a subcomponent of their extensive research framework rather than adopt evolutionary methods as the central thesis of their model. Numerous studies including Yu

Fang et al. [4], Chung et al. [5], and Wang et al. [6] focus on genetic algorithms and evolutionary programming as auxiliary tools to optimize the training and architecture construction of Convolutional Neural Networks (CNNs) or other types of neural networks. These trained neural networks, in turn, comprise the bulk of the algorithm that attempts to optimize the trading strategies.

Another study conducted by Zolfaghari et al. [3] delves into a Multi-Objective Genetic Algorithm, specifically focusing on its role in the feature selection aspect of predictive modeling. This research underscores the remarkable effectiveness of evolutionary algorithms in handling crucial tasks like feature selection. However, it also represents yet another instance where evolutionary techniques serve as integral subcomponents within a larger model. This study seeks to investigate the potential for Genetic Algorithms and Evolutionary Programming to serve as primary mechanisms driving the development and refinement of stock market trading strategies.

III. PROPOSED APPROACH

The utilization of independent Genetic Algorithms and Evolutionary Programming Algorithms in devising optimal intraday trading strategies stems from the need for adaptable, evolving models capable of navigating the complexities of the stock market. This proposed approach seeks to exploit the innate properties of GAs and EP to generate and refine trading strategies based on historical intraday stock market data, with a focus on maximizing profitability within a specified timeframe.

A. Data Selection

In this study, pre-selected stocks and historical data timelines were utilized as inputs to the model, employing 1-minute interval price data spanning from November 8, 2023, to November 28, 2023. The 20-day duration is chosen primarily due to constraints imposed by the availability of detailed intraday stock market data, as well as the computation time required to iterate through the enormous quantity of data. Utilizing this time frame also allows for a comprehensive analysis of short-term trends and volatilities within the chosen stocks' trading histories without sacrificing computational time. The tickers chosen for analysis in this research are shown in Table 1. These stocks were selected due to their high volatility and substantial daily trading volume.

AAPL	MARA	AMC	RIOT
IONQ	AFRM	AMZN	MSFT
CVNA	META	AMD	GOOG
MRO	DVN	PR	NVDA
ETSY	KLAC	ALGN	TER
BBWI	NOW	ADBE	CZR
PYPL	MPWR	TECH	MU
DHI	SNPS	INTU	CRM

Table 1. Selected Stocks

B. Representation

In both the GA and the EP-based approach for intraday stock market trading, each individual within the population is encoded as a list of floating-point variables. Each variable represents a different trading strategy, totaling to 26 floating-point numbers for each individual, allocating 13 numbers for buy strategies and 13 for sell strategies. Each floating-point number in the representation acts as a weight, determining the significance or likelihood of employing a specific strategy during the buying or selling phases in stock trading. These weights indicate the relative importance or preference assigned to each strategy, which in turn influences the decision-making process within the trading algorithm. By adjusting the weights through the evolutionary process, the algorithm can iteratively optimize and adapt the trading strategies to achieve improved performance and better adaptability to varying market conditions. This comprehensive set of techniques contains widely used technical indicators and candlestick patterns to equip the algorithm to navigate the stock market dynamics effectively. List 1 shows a comprehensive list of the strategies employed by the respective buy and sell components of the algorithm.

LIST 1
TECHNICAL TRADING STRATEGIES IMPLEMENTED IN THIS RESEARCH

BUY STRATEGIES:

- **Morning Star Candlestick Pattern**: A bullish reversal pattern formed by three candles: a long bearish candle, a small candle with a lower close and open, and a long bullish candle. Indicates a potential change from a downtrend to an uptrend.
- **Hammer Candlestick Pattern**: A single candlestick pattern with a small body near the top and a long lower wick. It suggests a potential bullish reversal after a decline.
- **Piercing Candlestick Pattern**: Comprises two candles, a bearish candle followed by a bullish candle that closes at least halfway up the prior candle's body. Signals a potential bullish reversal.
- **Inverted Hammer Candlestick Pattern**: A small body near the bottom and a long upper wick, signaling a potential bullish reversal.
- **Three White Soldiers Candlestick Pattern**: Consists of three consecutive long bullish candles with higher closes, indicating a strong reversal and potential continuation of an uptrend.

SELL STRATEGIES:

- **Shooting Star Candlestick Pattern**: A bearish reversal pattern characterized by a small body near the bottom and a long upper wick, suggesting a potential trend reversal from bullish to bearish.
- **Dark Cloud Cover Candlestick Pattern**: Formed by two candles, a bullish candle followed by a

bearish candle that opens above the prior candle's high and closes below its midpoint. Indicates a potential bearish reversal.

- **Evening Star Candlestick Pattern**: Comprises three candles: a bullish, a small-bodied, and a bearish candle. It signifies a potential reversal from an uptrend to a downtrend.
- **Hanging Man Candlestick Pattern**: Comprised of a small body near the top and a long lower wick, indicating a potential bearish reversal.
- **Three Black Crows Candlestick Pattern**: Consists of three consecutive long bearish candles with lower closes, signaling a strong reversal and potential continuation of a downtrend.

STRATEGIES FOR BOTH BUY AND SELL:

- **Moving Averages Crossover Strategy**: Uses the crossover of short-term moving average and long-term moving average to generate buy or sell signals when the shorter average crosses above or below the longer average, respectively.
- **RSI (Relative Strength Index) Indicator Strategy**: Identifies overbought or oversold conditions in the market to generate buy or sell signals based on the RSI values.
- **MACD (Moving Average Convergence Divergence) Indicator Strategy**: Uses the MACD line and signal line crossovers to identify potential buy or sell signals in the market.
- **Engulfing Candlestick Pattern**: Occurs when a larger candle completely engulfs the previous smaller candle, indicating a potential reversal.
- **Hamari Candlestick Pattern**: Involves a gap up or down on the third candle, indicating a stronger reversal.
- **Belt Hold Candlestick Pattern**: A single candle pattern with a small body near one end, signaling potential trend reversals.
- **Three Inside Up Candlestick Pattern**: A reversal pattern formed by three candles, where the third candle closes higher than the previous two.
- **Kicker Candlestick Pattern**: A two-candle reversal pattern where the second candle strongly gaps away from the previous candle's close, indicating a sudden change in sentiment.

The candlestick strategies integrated into this research were sourced from TA-LIB [9], a technical analysis library. I independently developed the Moving Averages, Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD) strategies. The specific parameters used in the moving averages strategy were a window size of 10 for the SMA (Simple Moving Average) and a window size of 50 for the LMA (Long Moving Average). The specific parameters used to create the MACD line from the MACD strategy were a short-term exponential moving average (ema) period of 12 and a long-term exponential moving average period of 26.

C. Crossover

Crossover facilitates a generation of new solutions by combining information from multiple parent solutions. The crossover operator chosen for the Genetic Algorithm was Whole Arithmetic Crossover, which works by taking the weighted average of parent solutions' genes with an alpha value to determine the proportion of genetic information inherited from each parent. This method ensures the preservation of valuable genetic material while also allowing for the exploration of novel solution spaces through a blending mechanism. The crossover operator performed in this study operated on 70 percent of the population with an alpha value of 0.75. While this crossover operator was exclusively employed within the Genetic Algorithm to generate diverse solutions by combining parental genetic information, the Evolutionary Programming algorithm relied on other evolutionary techniques for exploring and evolving solutions.

D. Mutation

Mutation serves as a crucial exploration mechanism to introduce diversity within the population of individuals. The mutation operators are employed differently between the Genetic Algorithm and the Evolutionary Programming Algorithm. The mutation operation in the Genetic Algorithm operates by adding a floating-point number to an element within an individual. This perturbation is carried out by sampling from a fixed Gaussian distribution. This mutation operation occurs with a 3 percent probability for every floating-point number in an individual. The mutation operation in the Evolutionary Programming algorithm operates more dynamically. The process works by adjusting the Gaussian distribution that employs a mutation step size that varies based on the observed changes in the mean fitness of the population. This mutation process is performed with a probability of 5 percent which allows the ES to dynamically fine tune the exploration-exploitation trade-off throughout the optimization process. Both the GA and EP utilize mutation strategies to diversify the population of trading strategies, discover potentially optimal solutions, and prevent premature convergence to suboptimal solutions.

E. Repair

A repair operator was employed after crossover and mutation were carried out in both the GA and EP algorithms. The goal of the repair operator is to ensure that the floating-point numbers representing the strategies of buy and sell actions conform to specified criterion. Because the individual is described as a list of percentages, these percentages must add up to 100% for both the buy and the sell sections of the individual. The repair operator acts in such a way to ensure that the relative weights among trading strategies remain equal.

F. Parent Selection

The process of selecting parent candidates in GA and EP algorithms is critical for the success of the optimization process. The parent selection operator that was chosen for the Genetic Algorithm is tournament selection. The process of tournament selection involves randomly selecting a subset of individuals from the population, and then choosing the individual with the highest fitness evaluations score from this subset to be a parent for reproduction. In the Genetic Algorithm, a tournament size of 3 was found to streamline the selection of the best-performing individuals in the population while still preserving diversity within the population, aiding in the exploration and exploitation of the search space. In the Evolutionary Programming algorithms, parents were selected deterministically as each parent produces one child through mutation.

G. Offspring Selection

The offspring selection decides which individuals will move on in the generation. The (μ, λ) Genetic Algorithm, straightforwardly replaces the current population with the newly generated offspring. Conversely, the (μ + λ) Genetic Algorithm employs elitism, combining parents and offspring to selectively form the next population from this combined list. In contrast, EP's offspring selection mechanism engages in pairwise competitions using a round-robin format. This involves evaluating each individual from the parent and offspring pools against a random selection of q counterparts (set to 5 in this research). For each q, a 'win' designation is awarded if the individual outperforms its opponent in each comparison. The solutions accruing the highest count of wins are then chosen to serve as parents for the subsequent generation. This meticulous method of offspring selection in EP harnesses competitive interactions to preserve and promote superior solutions, thereby driving the evolution of trading strategies towards enhanced performance.

H. Fitness Function

The fitness function of both the GA and EP aims to evaluate the effectiveness of the trading strategies produced. Each individual is assessed based on their respective trading strategy weights. The primary objective is to maximize cumulative profit regardless of the risk involved. Whether or not the stock is bought or sold is determined by aggregating the weights assigned to buy and sell strategies. A stock is considered for purchase if the collective weight of all buy strategies exceeds 0.1. A stock is similarly sold if the collective weight of all sell strategies exceeds 0.1. Because this algorithm is to be optimized for intraday trading, a constraint is placed on the algorithm which requires the stock to be sold within 20 periods. If the stock is not sold within this period, then it is automatically liquidated regardless of the stock’s price. Both the GA and ES are trained on the first 80% of the historical stock market data. After the algorithms finish running, the best-performing individuals from the population are forward tested on the remaining 20% of the data. This testing phase serves as an independent assessment to evaluate the real-world robustness and performance of the selected strategies.

IV. Experiments

This section explains the experimental setup, implementation details, and empirical results of the Genetic Algorithms and Evolutionary Programming Algorithm.

A. Experimental Overview

To conduct the experiment, historical one-minute stock data was collected from multiple stock tickers found in Table 1 using yahoo finance [7] for the period of November 8 to November 28. The stock data consisted of open, close, high, and low prices as well as volume traded for each one-minute period. The data was then processed, including normalization to ensure reliability of the dataset.

The experimentation initiated with the training phase using 80% of the available data. This training dataset comprised of 11 actual trading days, and the performance evaluation was conducted on three distinct algorithms: two variants of Genetic Algorithms (GAs) – one utilizing a (μ, λ) strategy replacing parents with their offspring, and another using a (μ + λ) strategy employing elitism in selecting the next generation; additionally, an Evolutionary Programming Algorithm was assessed under the same conditions. The assessment criteria involved measured the profit generated by each individual over the 11-day training period. The profit was quantified as a percentage for each individual over these 11 days and recorded as a decimal value. A positive value indicated that the algorithm generated profit, while a negative value indicated that the algorithm incurred losses during this training phase. Following the training phase, the entire final population from each experiment underwent further evaluation. These individuals underwent forward testing using the remaining 20% of the data, spanning a period of 3 trading days. During this forward testing phase, the profitability of the selected models was again measured in the same manner, calculating the profit percentages over the 3-day period. This forward testing served the purpose of assessing the effectiveness and robustness of the models developed during the training phase, evaluating their ability to perform on previously unseen data.

B. GA Implementation Details

The Genetic Algorithm is implemented using python, leveraging DEAPs (Distributed Evolutionary Algorithms in Python) [8] framework. This generational Genetic Algorithm is comprised of a population size of *i* individuals, each represented by 26 floating-point numbers, with 13 parameters allocated for buy trading strategies and another 13 for sell strategies. The algorithm iterates through *g* generations, applying a whole arithmetic crossover with a probability of 70% and introducing a Gaussian mutation to 3% of the genes in each individual. Selection of individuals for the next generation was accomplished via a tournament of size 3. In the (μ, λ) strategy only offspring are considered for selectin, whereas in the (μ + λ) strategy parents and offspring are considered. The algorithm's decision-making process occurs at each period within the dataset, where it determines whether to initiate a stock purchase, maintain the current position, or execute a sell order based on the trading strategies encoded in the individuals. The algorithm specifically tests the market's most volatile hours, from 9:30 am to 12:40 pm, capturing the period of highest trading volume while economizing on computational resources by excluding the latter part of the trading day. This algorithm maximizes the total profit accrued throughout the entire dataset period.

C. EP Implementation Details

The implementation of the Evolutionary Programming algorithm mirrors the structure of the Genetic Algorithm with a few notable distinctions. EP, akin to GA, utilizes a population-based approach with *i* individuals, each represented by 26 floating-point numbers encapsulating trading strategies and runs for *g* generations. However, a fundamental departure from GA lies in the absence of a crossover operator within EP, which relies solely on mutation for genetic variation. Mutation is applied to 5% of the genes within each individual and introduces an additional parameter known as the mutation step size. This step size plays a pivotal role in balancing exploitation and exploration tendencies by dynamically adjusting the Gaussian mutation distribution based on the ratio of the current mean fitness to the previous mean fitness. This adaptation facilitates the algorithm's ability to exploit promising solutions while exploring new potential areas in the search space. Apart from these differences, the training process for EP remains identical to that of GA, optimizing trading strategies to maximize total profit across the dataset period.

D. Results

The experimental evaluation of the (μ, λ) and (μ + λ) Genetic Algorithms and the Evolutionary Programming Algorithm unfolded intriguing insights into their performance. All experiments involving these algorithms were carried out on a 2.3 GHz Dual-Core Intel Core i5 processor. The first experiment run was to test all three of these algorithms on the same parameters of 15 generations for a population of 30 individuals. The computational time required for these evaluations were very similar ranging from 1 hour and 3 minutes to 1 hour and 14 minutes.

Figure 1 shows the evolution of the (μ, λ) Genetic Algorithm. The best individual from this population was able to achieve a fitness of 99.9% profitability. The evolution of this population shows a small, iterative increase in fitness at each point in the algorithm, without a notable sign of plateauing. This model clearly does not achieve convergence which suggests that a combination of a larger population and/or generations would achieve a better population of solutions. Figure 2 compares the forward tested fitness with the backward tested fitness the check the robustness of the model. It can clearly be seen from this figure that all solutions in the final population achieved a profit of 25% or higher, and the 4 solutions that performed the highest in back testing achieved a profit of 40% or greater in forward testing, supporting the idea that this algorithm produces profitable trading strategies.

Figure 3 shows the evolution of the (μ + λ) Genetic Algorithm. The best individual from this population was able to achieve a fitness of 96.58%. This algorithm reaches better solutions more quickly than the (μ, λ) strategy but sacrifices an early convergence which can clearly be seen in the converging maximum, minimum, and average fitness values of the population. Figure 4 compares the forward tested fitness with the backward tested fitness. The convergence is highlighted in this figure which only consists of a few unique individuals with forward fitness tested values of 45% to 53% profitability, showcasing a conclusively profitable strategy.

Figure 5 shows the progression of the Evolutionary Programming algorithm. The best individual from this population was able to achieve a fitness of 100.70%. A notable convergence of the maximum, minimum, and mean population was observed, and a steady upward trend is present. Figure 6 compares the forward tested fitness with the backward tested fitness. This figure shows that the final population achieves a decidedly profitable trading strategy, with 90% of individuals boasting fitness between 54% and 55%.

Next, the (μ, λ) Genetic Algorithm was trained with more computational resources consisting of a population of 50 and run for 75 generations to see if a better global optimum could be reached. The computational time required for this algorithm was 7 hours and 54 minutes, and the results of this Genetic Algorithm can be seen in Figure 7. This algorithm’s best individual achieved a fitness of 126.58% profit in the training stage, which is notably higher 99.9% profit from than the same model trained on a smaller population and generation size. The forward testing of this algorithm is demonstrated through Figure 8. The correlation of 0.34 between the backward and forward testing of individuals in this model was notably stronger than in previous experimental iterations, indicating increased consistency and reliability in predicting the algorithm’s performance in real-world scenarios. The individuals that tested the best in the backward tested data are all concentrated around the 40% to 44% profitability mark, indicating that these trading strategies are robustly profitable.

Next, the Evolutionary Programming Algorithm was trained with more computational resources consisting of a population of 75 and run for 50 generations to see if a better global optimum could be reached. The results of this experiment can be seen in Figure 9, where the best individual in this population achieved a profit of 112.91%. This is notably higher than the same model trained with less computational resources which produced a best individual of 100.7%. The forward testing of this algorithm, demonstrated through Figure 10, shows that a majority of the population achieved profits greater than 40%, with the individuals who performed the highest on the backward testing data achieving over 44% profit. Figure 10 shows a notably upward trend demonstrating that the individuals in the backward tested population have a more direct correlation of doing well in the forward tested population if they performed highly in the backward test.

Finally, experimental sanity tests were conducted to show that the strategies found in these algorithms tested better than traditional algorithmic trading methods. An experimental test of some traditional algorithmic trading strategies including the Moving Averages Crossover, RSI, and MACD trading strategies was conducted on the same set of forward tested data, yielding a profit of 7.86%, 4.44%, and 14.14%, respectively. Across the board, all Genetic Algorithms and Evolutionary Programming algorithms beat these numbers.

Next, all 26 trading strategies employed in this study were amalgamated and tested on forward testing data set. Through this conclusive examination, stocks were traded with a 100% adherence to their designated strategies, resulting in an impressive profit margin of 29.53%. Remarkably, among the multitude of strategies optimized through evolutionary methods, only a solitary individual in any final pool demonstrated a profit lower than this figure. Furthermore, a significant majority of these individuals from the GA and EP algorithms surpassed this benchmark by margins ranging from 10% to 25%. This definitive assessment unequivocally underscores the effectiveness of leveraging Genetic Algorithms and Evolutionary Programming Algorithms in identifying and refining optimal trading strategies.

E. Analysis

The experimental evaluation conducted in this study presents a comprehensive analysis of (μ, λ) Genetic Algorithms, (μ + λ) Genetic Algorithms, and Evolutionary Programming algorithms applied to the task of developing profitable trading strategies. The insights gleaned from these experiments shed light on their performance, convergence patterns, robustness, and comparative advantage against other traditional algorithmic trading methods.

One noteworthy aspect discerned throughout the exploration of these evolutionary algorithms is the consistent positive profit achieved by all individuals at various stages of the evolutionary process. This remarkable outcome underscores the effectiveness of the trading strategies devised within this study. The algorithms exhibited a collective proficiency in discerning opportune moments for transactions, adhering to a philosophy of buying and selling only when deemed necessary. This disciplined approach not only contributed to the profitability of the algorithms observed but also highlights the robustness the strategies employed by this study.

Initial experimentation demonstrated the (μ + λ) GA strategy converged the quickest, trailed by the EP strategy, while the (μ, λ) GA strategy showed little sings of convergence. In these experiments, all the best candidates achieved an impressive 40% to 54% profitability when forward tested, with the (μ + λ) GA strategy and the EP strategy producing the most promising solutions.

Further experimentation was conducted on the two algorithms that exhibited the slowest convergence during the initial testing phase. The (μ, λ) Genetic Algorithm strategy yielded a back-tested individual showcasing a remarkable 126.58% profit, while the Evolutionary Programming (EP) strategy produced an impressive back-tested individual with a 112.91% profit margin. Upon forward testing, both final populations demonstrated a consistent level of profitability compared to the initial experimentation but unveiled a notably pronounced correlation between scores observed in both forward and backward testing. This strongly implies a higher probability of superior performance within a real market setting for these individuals.

Next, the comparative analysis against traditional algorithmic trading methods provided the most compelling evidence of the superiority of both Genetic Algorithms and Evolutionary Programming. Across the board, these evolutionary algorithms outperformed traditional strategies such as Moving Averages Crossover, RSI, and MACD, showcasing their efficacy in generating profitable trading strategies.

The final conclusive test amalgamating all 26 trading strategies underscored the remarkable profitability achieved by the evolutionary optimized strategies. With only a negligible fraction of individuals failing to surpass the profit generated by this combined approach, the overwhelming majority significantly outperformed it, consolidating the effectiveness and superiority of Genetic Algorithms and Evolutionary Programming in devising optimal trading strategies.

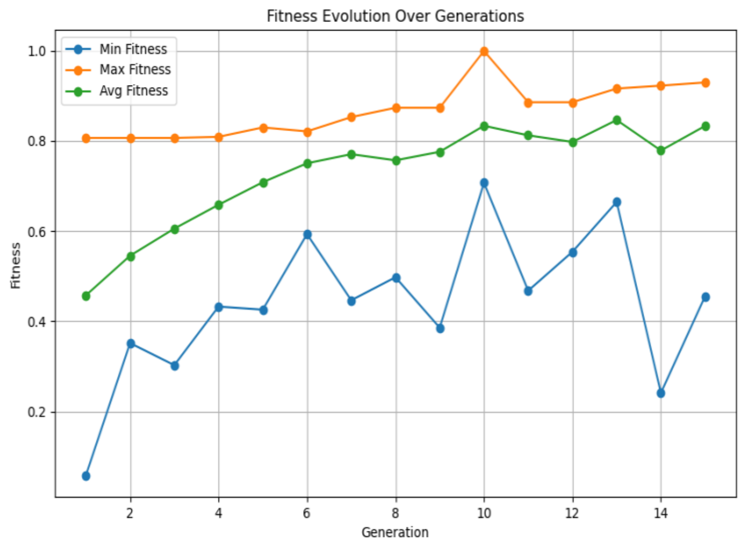

Figure 1

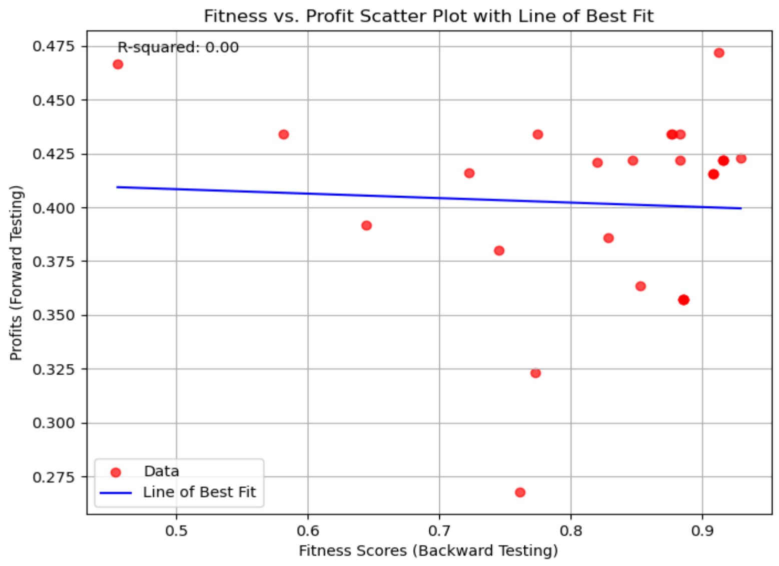

Figure 2

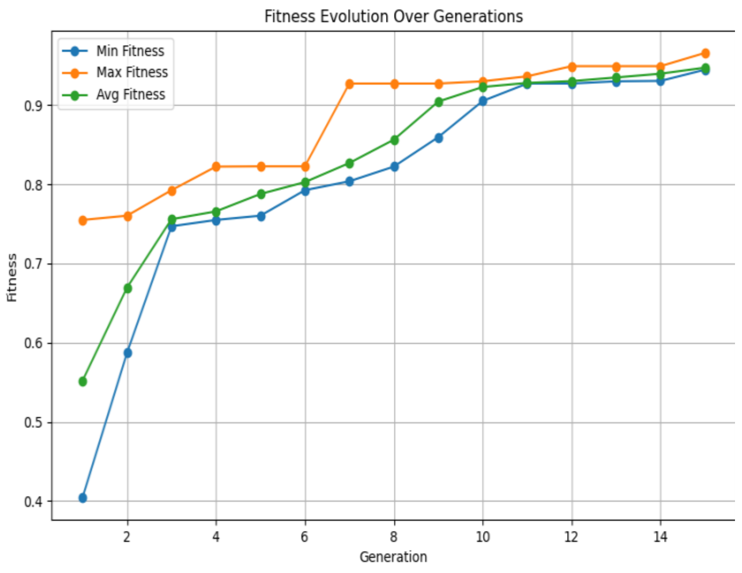

Figure 3

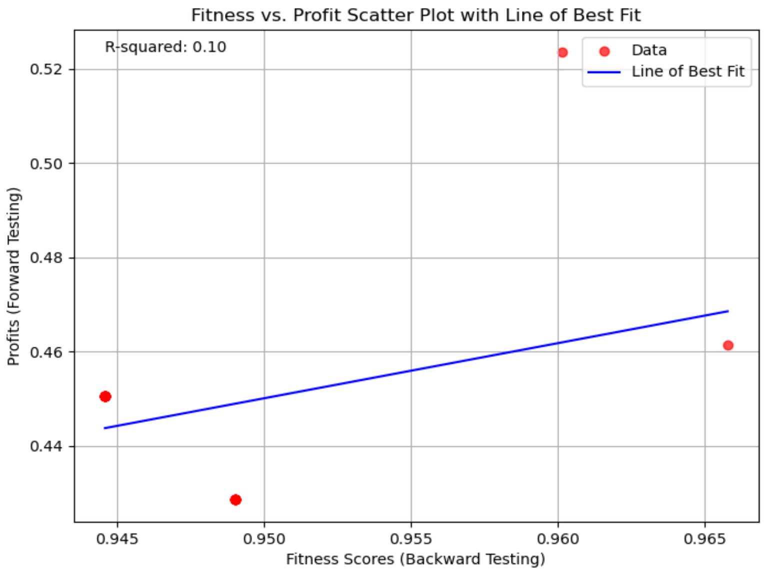

Figure 4

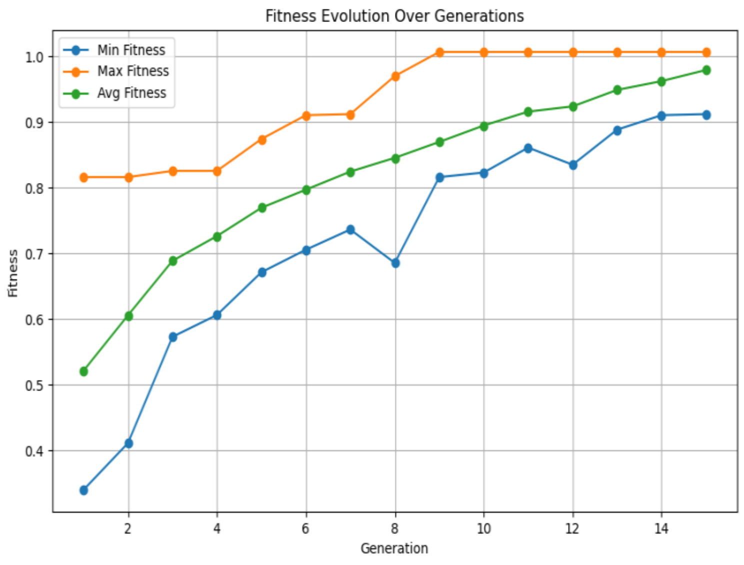

Figure 5

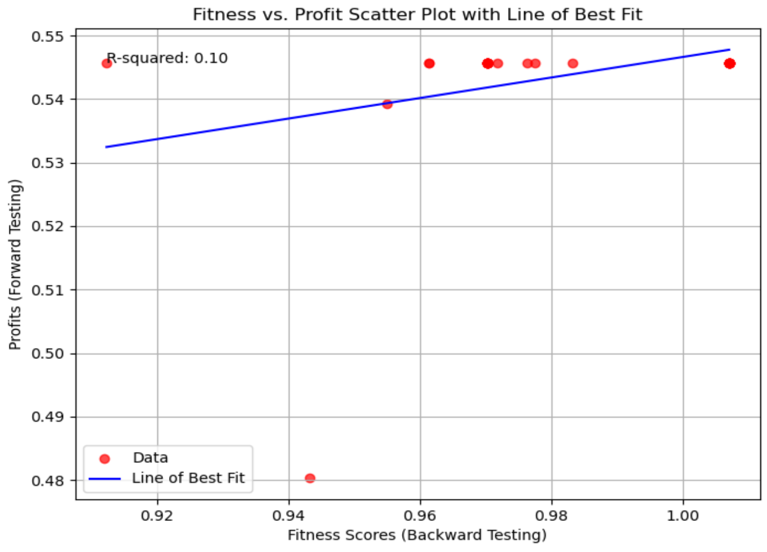

Figure 6

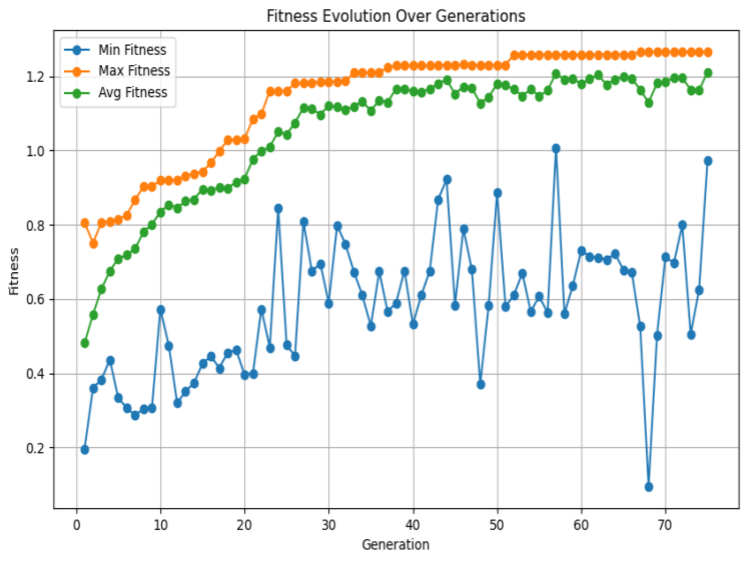

Figure 7

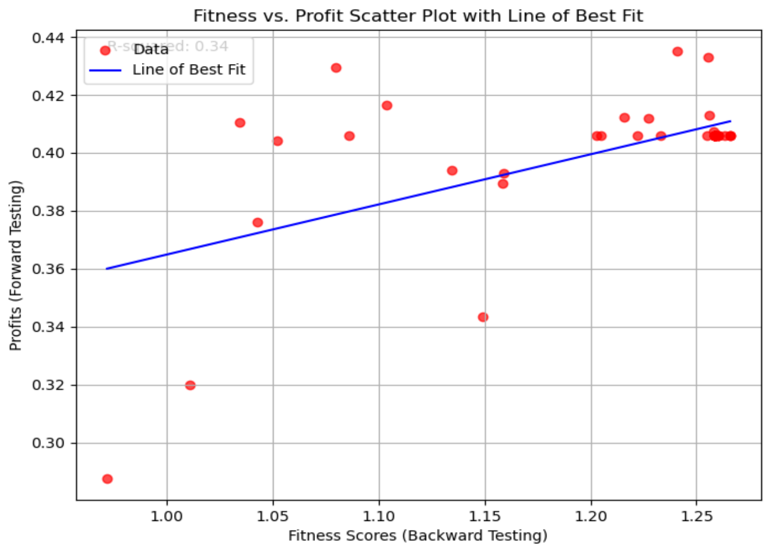

Figure 8

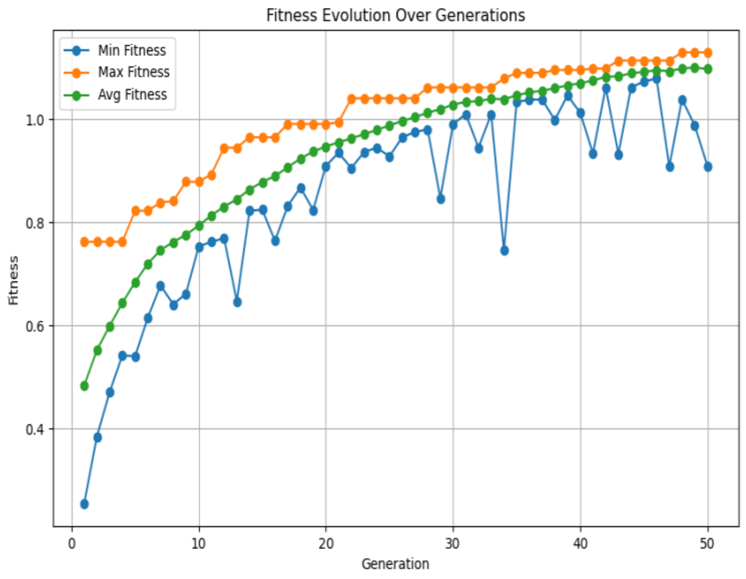

Figure 9

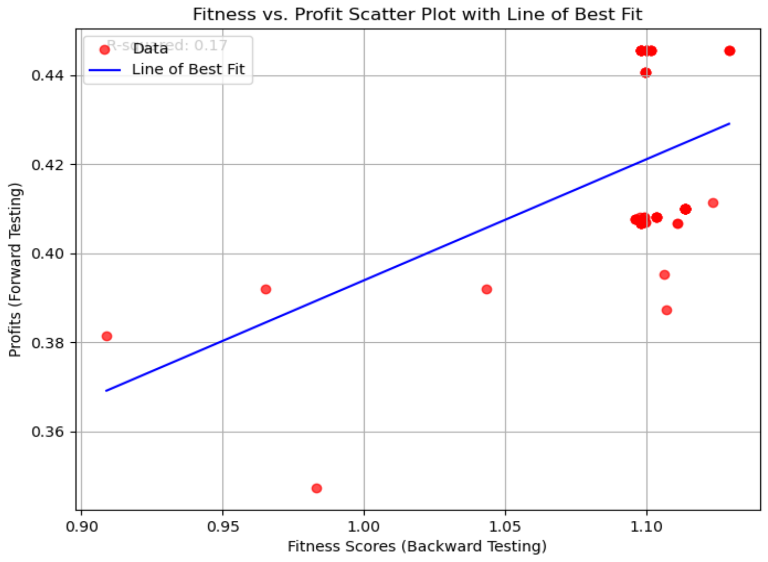

Figure 10

V. CONCLUSION

In this study, the application of Genetic Algorithms and Evolutionary Programming as primary engines for intraday trading strategy formulation and optimization reveals promising results. Experimental analyses conducted on a curated selection of stocks over a specific timeframe demonstrate the efficacy of these evolutionary computation techniques in generating profitable trading strategies especially when compared to traditional algorithmic trading methods. However, the research also opens avenues for future exploration. Transitioning these findings into live market simulations, where prices fluctuate at millisecond intervals, offers a crucial step towards validating and fine-tuning these strategies in real-time environments. Other steps that can be taken to expand on this research would be to access more extensive historical data, particularly at a more granular level, such as tick data, which may be able to provide deeper insights into market behaviors and aid in refining the strategies further. Additionally, it would be interesting to incorporate a larger selection of trading strategies into these algorithms such as Bollinger Bands to potentially improve the model. Moreover, expanding the repertoire of Evolutionary Programming and Evolutionary Strategies by incorporating additional self-mutating parameters may enhance the adaptability and robustness of these algorithms. Overall, this research offers promising avenues for advancing the domain of algorithmic trading using evolutionary computation techniques.

REFERENCES

[1] Badawy, F.A., Abdelazim, H.Y. and Darwish, M.G. (2005) 'Genetic Algorithms for Predicting the Egyptian Stock Market', 2005 International Conference on Information and Communication Technology, Information and Communications Technology, 2005. Enabling Technologies for the New Knowledge Society: ITI 3rd International Conference on, Information and Communications Technology, pp. 109–122. doi:10.1109/ITICT.2005.1609619.
[2] Zolfaghari, M. et al. (2022) 'Stock Market Prediction Using Multi-Objective Optimization', 2022 12th International Conference on Computer and Knowledge Engineering (ICCKE), Computer and Knowledge Engineering (ICCKE), 2022 12th International Conference on, pp. 253–262. doi:10.1109/ICCKE57176.2022.9960002.
[3] Drake, Adrian & Marks, Robert. (1998). Genetic Algorithms In Economics and Finance: Forecasting Stock Market Prices And Foreign Exchange — A Review. 10.1007/978-1-4615-0835-9_2.
[4] Yu Fang et al. (2014) 'Improving the genetic-algorithm-optimized wavelet neural network for stock market prediction', 2014 International Joint Conference on Neural Networks (IJCNN), Neural Networks (IJCNN), 2014 International Joint Conference on, pp. 3038–3042. doi:10.1109/IJCNN.2014.6889969.
[5] Chung, H. and Shin, K. (2020) 'Genetic algorithm-optimized multi-channel convolutional neural network for stock market prediction', Neural Computing & Applications, 32(12), pp. 7897–7914. doi:10.1007/s00521-019-04236-3.
[6] Wang, H. (2018) 'A Study on the Stock Market Prediction Based on Genetic Neural Network', Proceedings of the 2018 International Conference on Information Hiding and Image Processing, pp. 105–108. doi:10.1145/3292425.3292441.
[7] Yahoo Finance API (2023) Historical Data for AAPL using Yahoo Finance API. [online] Available at: https://finance.yahoo.com/market-data/ (Accessed 29 November 2023).
[8] Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M., & Gagné, C. (2012). DEAP: Evolutionary Algorithms Made Easy. Journal of Machine Learning Research, 13, 2171–2175.
[9] Benediktsson, J. ta-lib-python [Computer software].