Game Theoretical Approach to Stock Selection using Fuzzy Logic and Genetic Algorithm

Amay Yadav

EL-GY 9213 Game Theory

New York University Polytechnic School of Engineering
NYC, USA

amay@nyu.edu

***Abstract*—— this is a comprehensive study of three different papers listed in the bibliography. I have read them thoroughly and all of them have a basic approach to stock portfolio selection first an algorithm is run to judge the score of a stock. This algorithm is not a conventional algorithm by any means and this score is also not the value of a stock but a dataset that tells us its relative grading with the other stocks. The two most popular methods used for scoring these stocks are fuzzy logic and genetic algorithms. These two methods are many-valued logic, they are very different from one another but the basic concept on which they run are the same which is to give a better output after every time they run. They also may give different outputs every other day. After retrieving the scores from these algorithms we can apply them in our trading strategies which are made up based on concepts of game theory.**

The game theory concept mostly used in these papers is the zero-sum game. As all the investors and brokerage firms are investors and henceforth the players and the stocks net asset value is the payoff which can lead them to profit or loss. Hence it is mathematical representation of a scenario wherein every player loses or gains utility but at the end of the day the total sum of all the gains are subtracted from the total losses they will be equal to zero, thus the market remains in a state of equilibrium. It's a repeated zero-sum game to be precise and every player strives to profit.

***Keywords*—— stock selection; repeated games; payoff utility, zero-sum games; fuzzy models and membership function; genetic algorithm.**

I. Introduction

Financiers and investors are always looking for a good investment. There are Exabyte's of data out there concerning the stock market and there are a lot of ways to judge if a stock is going up or down. At the end of the day it will always be upon the investors understanding capability and mainly speculation which governs there decision on which stock to buy. In a real world application there are unknown scenarios which a computer algorithm cannot evaluate in a static or sequential manner especially the stock selection process. Hence, I have picked up papers which apply dynamic programming and soft computing logic for stock selection using a game theoretical approach.

We can compute stock data using different methods and parameter functions (historical data, news data, empirical data, yield data etc.) to maximize our return or in better words expected return. Now the stock trading market is just like any game theory Game, it has various players, choices, pay-offs, threshold loss (risk), equilibrium, security levels and its stochastic nature. The data up to the present date and time is complete information but the selection of stock will be done in expected result value which is an incomplete information game. Even the time matters i.e. when to buy what stock and when to sell a stock.

This game is played using fuzzy logic and Genetic Algorithm to evaluate an expected profit on the stocks. First a score is obtained for the stocks using their function parameter and fundamental variables. These scores can then be used to relatives scale the stocks. We can then make portfolios of the selected stocks i.e. the high scoring stocks.

Fuzzy logic is an excellent way to evaluate and rank stock data because it is imprecise or approximate, complex and it requires reasoning. Thus the entire stock data other than the empirical stock values can be included in a fuzzy set and used in a ranking system.

Genetic Algorithm is used for further optimization of the stock ranking. Genetic algorithm is based on the concept that the next generation will be better than the last after mutation. Which means it is a perfect way to judge something so random because it is dependent upon the dynamic data itself.
Game theoretical approach for any stock trading strategy or evaluation is what Gordon Gecko (Wall Street (1987)) coins very accurately "It's not a question of enough, pal. It's a zero-sum game." In game theory or economic theory zero-sum games are those in which the payoff of any player is exactly balanced to the other player's loss.

Game theoretical approach is applied by many investment firms all across the globe and to understand those strategies we need to understand there school of thought. Now any stock or portfolio selection requires copious amounts of data and statistical and mathematical analyses. After all those calculations are carried out the final stage of implementation boils down to three things buying, holding and selling.

Now to understand these strategies we can classify them into the Fundamental Approach and Technical Approach. The fundamental approach is the one which involves careful analysis of the company's books; studying the company's assets and its expenditure and income. It also includes the careful evaluation of the news and the mindset of all the big brokers (players) in the business. It also depends upon the proposed future of the company and its upcoming products. [1]

The Technical Approach states that we can predict a company's future stock price by studying the company's past. Back in the day the technical approach was not steady or constant and so people dismissed it. After the emergence of technology and better computers, Traders can now use rigorous data mining techniques to implement the technical approach and get better results. [1]

To prove there theory was actually in an implementation stage the authors looked through a lot of data about the research done before them and they found that Allen and Karjalainen had already worked on their proposed theory before and they found their research quite insightful. Allen and Karjalainen had taken a training data set and worked out a genetic algorithm which gave a worthwhile decision of 'buy' or 'sell'. This algorithm was run for a finite amount of days and the results led to higher absolute returns. Some more work was done by Protvin but that didn't lead to beneficial result. These algorithms worked only when the market was in a flurry and they didn't exactly give a good results when the market was stable. [1]

Real headway was made by Dourra and Siy, who used fuzzy logic to create a trading strategy. Their fuzzy logic system took inputs that were momentum indicators (derived from the deviation of different from a changing statistical value), such as the rate of change of stock prices over a defined period. The Fuzzy rule base was based on Gaussian membership functions. This fuzzy logic system is fed into a neural network and not just any neural net but the Mamdani neural network. The system would output strong sell '0' or strong buy '100'. It was run on real world data of 3 year stock process for different companies. It gave 250% profit margin on its investment. The high yield was due to effectiveness of the algorithm and key adjustments were made in the algorithm to suit the given training data. [1]

The final study that was observed was made by Lam who also made fuzzy trading rule but he added an adjunct that is the genetic algorithm thus rendering the program as it is coined "co-evolutionary". The fuzzy rules are significant for this process and to understand how the data is actually being processed I am quoting the author "The antecedent of each rule was a conjunction of several moving average trends such as daily moving average, weighted moving average and exponential moving average. Other fuzzy rules had antecedents with momentum terms such as relative strength index, rate-of-change and fast and slow stochastic." This fuzzy logic was passed to neural network called the Sugeno method of fuzzy inference which trains itself by shifting weights in the input data. The output of this particular model is unclear and so is its training but it is useful to recognize the potential for evolving fuzzy trading rule-bases. [1]

The goal of this research is to create agent-based stock market games that are generally nonlinear in state and non-quadratic in objective (i.e. profit function). The authors have used a stochastic dynamic stock price model that are affected by the players' actions in the market which are quite different from the traditional time-series models for stock process that are solely based on historical data. They have also formulated non-quadratic functions for the games to maximize the player's profit. The logical soft computing for the non-quadratic functions is done using gradient-based and direct search method which are evolutionary in nature. The chosen method is obviously the genetic algorithm (GA) for solving our stock-market game and finding the optimal strategies for the agents at each trading period of the market. [2]

Now the data for any historical market needs to be mined and classified that is for it to run the fuzzy logic and genetic algorithm it needs to converted into a more refined form and that is a Big Data Problem which is carried out by various techniques of machine learning. Some of the machine learning techniques used are the ANN's (Artificial Neural Network) or the SVM's (Support Vector Machines), Evolutionary algorithms (EA's) and fuzzy inference models. Some people have already trained neural networks for estimation and prediction of asset behaviour in order to facilitate decision-making in asset allocation. Although these models worked in some applications, they often suffer from the over fitting problem and may tend to fall into a local optimum. [3]

The paper states that previous fuzzy logic schemes for stock selection lacked the learning ability and that they didn't result in profitable outputs. However, more recently a study was made on a hybrid neuro-fuzzy inference system for the forecast of financial time series. Chang showed that this hybrid model is able to improve the predictive accuracy of irregular non-periodic short-term time series forecast. This hybrid neuro-fuzzy inference system is exactly like the co-evolutionary algorithm as mentioned in the previous papers but one thing that it lacks is it doesn't take game theory into consideration. It also certain kind of input variables for it to work. As mentioned earlier this system works on machine learning techniques and one of the greatest problems with that is the classifier; for that to work efficiently we need to feed the input as specific patterns. Once the classifier, reaches the said classification accuracy for the learned neural network, then the computational overhead required for learning a classification function is directly proportional to the number of training examples needed for learning, and the cost associated with the features. Therefore, the goal of feature selection aims to identify useful, non-redundant subsets of features for a given data mining or machine learning task. [3]

The input variables are not just raw stock data but they also include learning models which usually consist of not only the features but also the models parameters. Therefore, the authors in this study have devises a hybrid fuzzy-GA stock selection model for this task, where the GA is used for feature selection and optimization of parameters, simultaneously. In a nutshell, the authors have created a stock scoring mechanism using fuzzy membership functions with the GA-optimized model parameters. Based on the scores calculated, top-ranked stocks are then chosen for portfolio construction. These stocks are believed to be beneficial investment over the long period of time. [3]

Hence it all boils down to an algorithm that works in a real world environment on a very stochastic and dynamic dataset. It applies soft computing logic and game theoretical logic to solve a mathematical problem.

II. GAME THEORY

Brokerage firms, develop trading strategies for stock market investments. They are essentially the players. Good trading strategies attract more investor dollars and make a huge profit which is the gain in payoff utility function while, poor strategies bleed out money and take huge losses which is the loss in payoff utility function. Each firm develops multiple investment strategies to deal with the stochastic and unprecedented nature of the market. For example, the firm will need one strategy to deal with investments in a bull market, where stock prices generally rise, and another strategy for bear markets, where stock prices generally fall. Investors individually choose the most profitable strategies and apply them in an ordered manner to gain the highest payoff. Thus looking at it from a game theoretical viewpoint we have laid forward a game where the players are brokerage firms and money is the payoff. [1]

This entire process can be envisioned as an evolving set of strategies where fitness (genetic algorithm output) is measured by the profits reaped by using the player's individual strategy. The golden question here is how do we judge if any of these strategies is good or not? [1]

That question illustrates the problem with evolving a single population of trading rules. Profitable investment trading strategies are directly proportional to high fitness function. But defining fitness proportional to returns is quite straightforward and may give a false picture of how good a strategy actually is. An example that I have directly picked up from the paper will help illustrate the problem. Suppose two brokerage firms independently evolve a set of investment strategies. Assume the best performing strategy from the first brokerage firm consistently yields a 4% return over a 90-day trading period. If fitness is proportional to returns, then this strategy, by definition, has the highest fitness. But does a 4% return really qualify as high fitness? The only way to know for certain is to compare that 4% return against what the other brokerage firm can offer. If the best strategy from the second firm only has a 1.5% return, then 4% is pretty good and should signify high fitness. On the other hand if the second firm can offer a 7% return strategy, then a 4% return does not qualify as high fitness. It is important to differentiate between the relative fitness and the true fitness of an investment strategy. Relative fitness compares returns from strategies within a single brokerage firm whereas true fitness contrasts returns from strategies between independent firms. Investors compare the returns achieved by various brokerage firms before deciding where to invest their money. Hence, true fitness measures the ability to attract investor dollars. Survival, therefore, should depend on true fitness rather than relative fitness. [1]

The point here is there is no way to say whether or not a given return constitutes high fitness unless it is compared against the returns from competing brokerage firms. To consider this problem in a logical manner the Stock market investment is expressed as a game. The players are brokerage firms, which individually develop investment strategies. It is these strategies that compete against each other in the marketplace and act like a rule-base for the players and the players receive payoffs in the form of money. Bad strategies lose money as investors switch to better performing strategies. Hence, stock market investment is essentially a zero-sum game with the strategies formed through competitive coevolution. In some games players make decisions based on historical data, which is common knowledge. One of the best examples is the widely studied minority game. The stock market game formulated here belongs to this same family of games. [1]

Game theory has various definitions but one thing is apt from all those definitions and it is not a mathematical construct but a financial one. The following context has been taken from the paper itself because it is a scenario defined by the author and it is imperative here to hear the author's words. "Game theory can help us in modelling the financial, economical, and multi-agent systems that are interactive; and the objective functions of each player and the system constraints are affected by other players' actions (or decision variables). A stock-market has the characteristics of a game in the sense that each stock market-player has the objective of maximizing its expected profit in the market by interaction with other players; and also each player's actions (i.e., selling, buying or holding a stock in a large volume) could affect the price of a stock in the market. The authors of the paper have modelled the stock market as a non-cooperative repeated (dynamic) game with four players (agents) who compete with each other to maximize their expected profits/wealth. The authors have picked a scenario with There are eight stocks to be traded in this market and two scenarios are simulated: 1) all four-players trade with each other in a sequential zero-sum game, 2) each player trades with the stock broker in a non-sequential non-zero sum game. The stock price changes at each iteration based on the players' actions in the market. This stock price model is based on the basic demand and supply theory." [2]

III. FUZZY LOGIC

Fuzzy logic is a very apt way to select stocks or predict their future prices for a stochastic process such as the stock market. Fuzzy logic is a form of many-valued logic; it deals with reasoning that is approximate rather than fixed and exact. Compared to traditional binary sets (where variables may take on true or false values), fuzzy logic variables may have a truth value that ranges in degree between 0 and 1. Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false. Furthermore, when linguistic variables are used, these degrees may be managed by specific functions. The fuzzy logic system follow the concept of fuzzy sets and not traditional sets which allow a level of diversity i.e. fuzzy functions don't always lead to a complete yes or no answer.

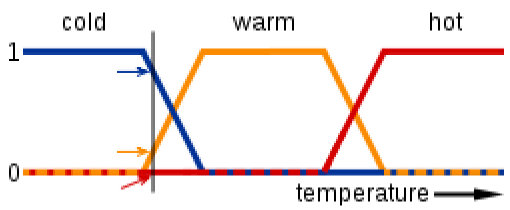


Fig. 1. Simplest fuzzy logic Trapezoids characterizing an AC

Now there is loads of data for a stock or a company that we can consider but the most prudent one is the historical stock prices, its deviation and pattern. The historical data of the stock market can be easily analysed but a more subtle and challenging problem must be dealt with. The rule-base contains a set of fuzzy rules that predict whether the next trading day is likely to be a good trend day. Investment decisions which are whether to buy, sell or hold rely on these predictions. Fuzzy rules are used because crisp rules are too restrictive. Crisp rules are traditional and only have a yes or no answer but the fuzzy rule gives you the probability of the event and it depends upon the investor's discretion to proceed forward or not. An example that I have picked up from the paper will explain this better. The crisp rule "if NR7 then ... " is true if and only if the range during the current day range is less than the range during any of the previous six days ranges. The rule won't be true if even one of the previous ranges is less than current day range. But suppose the inequality is satisfied for say five out of the six days, which makes the antecedent almost true.[1]

I have picked up another example from the paper to show the effectiveness of the fuzzy reasoning the author asked "How do you decide whether a near-DOJI day (that is, where the open and close are very close, but not exact) should be considered a DOJI? This is subjective and there are no rigid rules.... "Fuzzy reasoning can effectively deal with such uncertainties. Crisp rules, which must give either a 'yes' or a 'no' answer, cannot handle these situations but fuzzy rules can because they can also provide fuzzy answers somewhere in between a 'yes' or 'no'. [1]

The author has used the fuzzy logic approach to analyse the stock market data is analysed to determine how closely it matches the formal definitions of the prescribed features. The Membership functions of the fuzzy logic return a value between 0 and 1 indicating to what degree features are present. The resultant fuzzy variables are then collected into fuzzy if-then rules, which constitutes the trading rule-base. The outputs of these rules are combined into a fuzzy output variable. This variable is defuzzified to produce a crisp value on the unit interval, which the desirability of buying stock on the next trading day. [1]

IV. GENETIC ALGORITHM

In the field of artificial intelligence, a genetic algorithm (GA) is a search heuristic that replicates the process of natural selection. It is used to generate useful solutions to optimization problems .Genetic algorithms generates solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover.

Genetic Algorithm is a computational model inspired by evolution. I couldn't describe it any better so I have picked up the authors words "Genetic Algorithm has been introduced as an optimization method with abilities to solve non-linear and non-quadratic optimization problems. The GA is a discrete optimization method that is based on the fitness of an individual (or genome) in its environment (or feasible search space). The fittest individual will survive and move to the next generation. The main modules of the GA algorithm are: 1) population, 2) fitness evaluation, and 3) reproduction. The population size is a function of the size of the search space (the bigger the search space, the larger the population size) and it should be initialized at the beginning of the search. The fitness function is the objective function of an individual that should be evaluated in each generation. The reproduction is how the next generation of genomes will be created to continue the search for finding the best fitness value. The reproduction process is done three ways: 1) Elite individuals (or children) go to the next generation without any changes, 2) Mutation, where some of the genes of a parent genome are changed to create the child genome for the next generation, and 3) Cross-over, where the genes of two parent genomes are combined to create the child genome for the next generation."[2]

The author has applied GA algorithm for solving nonlinear stochastic stock-market games. Each player maximizes its expected profit (or fitness) function using the GA algorithm. The population size and the number of stocks considered are small for experimental reasons. There are 20 players and they trade on 8 different stocks. The author has used the prominent feature of the GA Algorithms which is the selection of elite children by mutation and cross-over to create the next generations and the author has defined an escape condition in this infinite loop by comparing the results of the current generation with the two previous ones and if the difference is 0.1 or less then the iteration stops. [2]

The basic and most humanized explanation of the Genetic algorithms is that they are models of natural evolutionary systems and as adaptive algorithms for solving optimization problems. Its application on computers is done by following the certain scenario where agent is comprised of a genotype (often a binary string) encoding a solution to some problem and a phenotype (the solution itself).GAs regularly start with a population of randomly generated agents within which solution candidates are embedded. In each iteration, a new generation is created by applying variations, such as crossover and mutation, to promising candidates selected according to probabilities biased in favour of the relatively fit agents. As a result, evolution occurs by iterated stochastic variation of genotypes, and selection of the best phenotypes in an environment according to how well the respective solution solves a problem (or problem-specific fitness function). Successive generations are created in the same manner until a well-defined termination criterion is met. The core of this class of algorithms lies in the production of new genetic structures along the course of evolution, thereby providing innovations to solutions for the problem at hand. The steps of a simple GA are shown in the following:

Step 1: Randomly generate an initial population of l agents, each being an n-bit genotype (chromosome).
Step 2: Evaluate each agent's fitness.
Step 3: Repeat until l offspring have been created.
a) Select a pair of parents for mating;
b) Apply variation operators (crossover and mutation);
Step 4: Replace the current population with the new population.
Step 5: Go to Step 2 until terminating condition.

The most fundamental parts of the genetic algorithm is the chromosome coding. Just think of it in terms of a stock encoding theme which takes in a lot of numbers and variables like historical data, deviation current trends and embeds it into single unit called the chromosome which is equivalent to a variable in the computer. The other fundamental part of the genetic algorithm is fitness function which can just tell you what that chromosome is worth and that worth can help you identify if that chromosome is good enough or not to survive. Chromosome encoding and fitness function Among many paradigms of search algorithms GAs have been proven to have an advantage over traditional optimization methods in problems with many complex, discontinuous constraints in the search space. Since the Genetic Algorithm is an evolutionary algorithm it can be designed to optimize various combinations of sets of features that improve given optimization criteria, this class of algorithms is better suited to cope with the feature interaction problems. Therefore, we propose to use the GA method to search for optimal subsets of features for the stock selection model. [3]

V. PROBLEM FORMULATION

It is hard to begin evaluating any stock data; so the authors start off by taking a single stock and recording its parameters for a large number of days. We need to predict the stock price at any given day so the author has kept the formula and the parameters simple. The result that we get is also made simple by declaring it into two categories up-trend day and down-trend day. These trend days are formulated using the following parameters: open share price (0), the high (H), low (L) and the closing share price (C) for any given day. [1]

Up-trend day

$$O <= L + 0.1(H - L)$$

$$C >=H-0.2(H-L)$$

Down-trend day

$$O >= H - 0.1(H - L)$$

$$C <=L + 0.2(H - L)$$

Mathematically, these derivations tell us that for an up day the opening price is close to the day's low and the closing price is close to the day's high and vice-versa for a down-trend day the opening is near the high and the close is near the low for the day. An interesting property of trend days-which keen investors can exploit-is the high-low differential tends to be relatively large.
some of the stock properties or parameters that are fuzzified are:
NRk – it is the volatile range of a particular stock, it calculated by the difference of high and low price of a particular stock at any given day. The range is defined as R = H - L. NRk exists if today's range is less than the ranges for the previous k - 1 days. That is, NRk days represent volatility contraction, which often times leads to volatility expansion in the form of wide range days. The greater the number of narrow range days, the greater the counter reaction in wide ranging days.

DOJI- it indicates that the open and close for the trading day are within some small percentage of each other. A DOJI means the market reflects temporary price indecision and often signals a major reversal in the market. It returns 1 (TRUE) or 0 (FALSE).

Hook day - A hook day occurs when the price opens outside the previous day's range and then proceeds to reverse direction, generally indicating a reaction to temporarily overbought or oversold market conditions. There are two versions of a hook day. One is the down-hook day it is the one in which the opening price of a particular stock is higher than its previous day's high (H) and the second is the vice-versa called the up-hook day it is the one in which opening price of a particular stock is lower than its previous day's low (L) [1]

The aforementioned parameters are then fuzzified. Fuzzification is the process that maps days (D) onto the unit interval via a membership function. In layman's terms D represents the number of days in which a particular parameter is sustained be a particular stock. For instance, for the hook-day membership function is equal to zero then that criteria is not met for any of the days in the fuzzy crisp date value set. [1]
Now to get a clear understanding of these fuzzified values the authors have drawn out linear inequality equations and solved them using the fuzzy trapezoidal membership functions.

$$\mu_k(x) = \begin{cases} 0 & x < \upsilon_{\min} \\ c\,(x - \upsilon_{\min}) & \upsilon_{\min} \le x < \upsilon_{\max} \\ 1 & x \ge \upsilon_{\max} \end{cases}$$

with parameter values as shown in Table I.

k	c	$\upsilon_{\min}$	$\upsilon_{\max}$
4	1/2	2	4
6	1/3	3	6
7	1/3	4	7

TABLE I

PARAMETER VALUES FOR NARROW RANGE FEATURES

Thus the trapezoid for the membership function for the features is as follows:

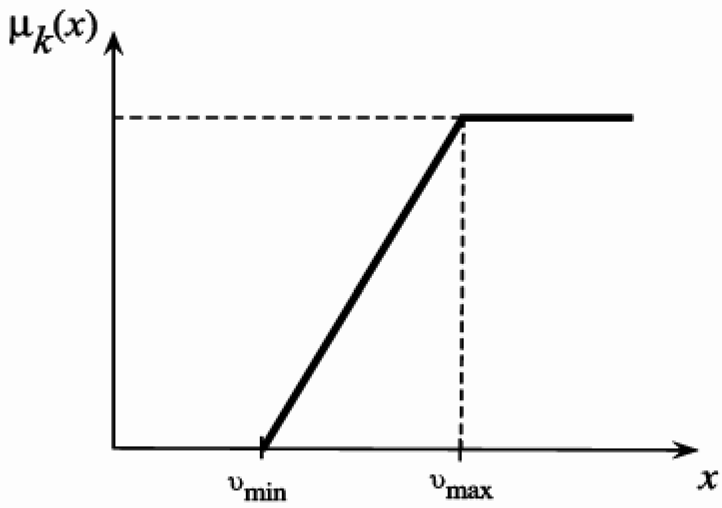


Fig. 1. Membership functions for the features.

The authors have then also carried out the same process with DOJI and Hook-days.
The membership function for the DOJI is:

$$\mu(x) = \begin{cases} 1 - x/\rho & 0 \le x \le \rho \\ 0 & \text{otherwise} \end{cases}$$

Here the p is the threshold parameter and it lies between 0.05 and 0.3. x is small percentage diffenrce betwwen the O and C. The trapezoid is plotted using p = 0.1.

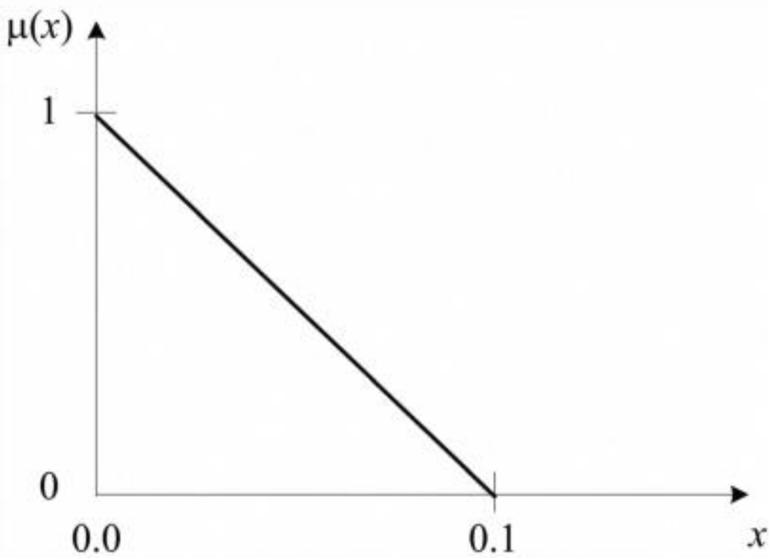


Fig. 2. Plot for the DOJI with p = 0.1

Hook day has also been applied to the trapezoidal function.

$$\mu(x) = \begin{cases} 0 & x < -\frac{1}{2} \\ 2(x + 0.5) & -\frac{1}{2} \le x < 0 \\ 1 & x \ge 0 \end{cases}$$

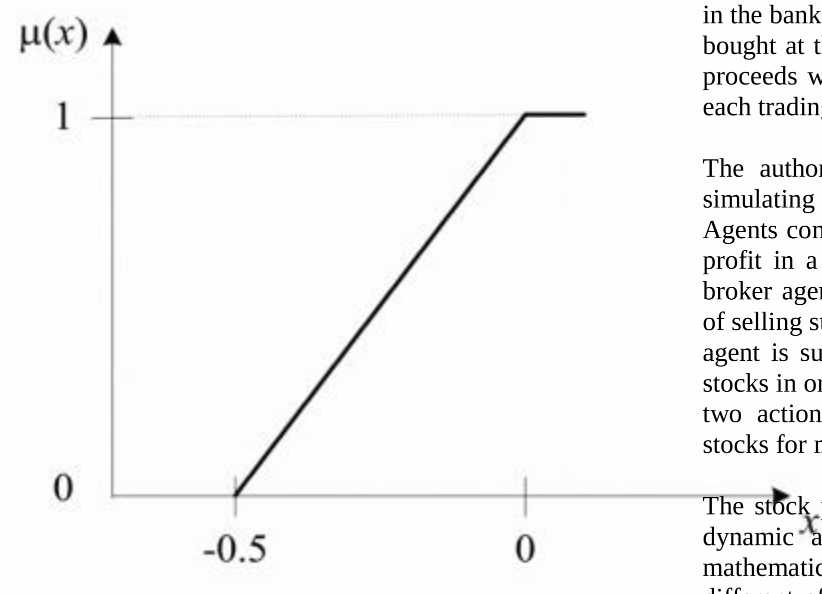


Fig. 2. Plot for the Hook-day

Once these trends are formulated using the fuzzy trapezoidal method we also need to pass them through a rule base. This rule-base is dependent upon the outputs, which means that if n number of features output m number of fuzzified outputs then there must be m number of rules. A singleton fuzzier is used for the inputs. The set of possible outputs is:

$$A = (\mu_{\mathrm{NR4}}, \mu_{\mathrm{NR6}}, \mu_{\mathrm{NR7}}, \mu_{\mathrm{DOJI}}, \mu_{\mathrm{hook}})$$

Using these outputs a matrix is formed which looks like

$$M = \begin{pmatrix} 0 & 0.6 & 0.5 & 0 \\ 0 & 0.33 & 0.33 & 0.33 \\ 0 & 0.1 & 1.0 & 0.9 \\ 0 & 0.44 & 0.5 & 0.1 \\ 0 & 0.1 & 0.2 & 0.7 \end{pmatrix}$$

Then fuzzy decision making is done using A o M = B which is something quite similar to the maxmin theory in game theory. [1]

$$b_j = \max_{1 \le i \le 5} \min \{a_i, m_{ij}\}$$

The fuzzy output vector B is then defuzzified to get a crisp output value indicating the desirability of buying shares of stock. A centre of average defuzzification was used. That is,

$$U = \frac{\sum_{i=1}^{N} b_i \cdot \lambda_i}{\sum_{i=1}^{N} \lambda_i}$$

The author has chosen 0.8for this threshold. If the output value is greater than 0.8, then a 'buy' signal is generated and how high above 0.8 determines the number of shares to buy. Specifically, the amount of stock purchased increased linearly the higher the output was above 0.8, subject to sufficient funds in the bank account, up to a maximum of 20 shares. Stock was bought at the opening price and sold at the closing price. All proceeds were deposited into the bank account at the end of each trading day.[1]

The author has formulated a repeated zero-sum game by simulating the genetic algorithm with two trading scenarios: 1) Agents compete with one another to maximize their expected profit in a stock market, 2) Each agent trades with a stock broker agent. The stock broker always creates the possibility of selling stocks to and/or buying stocks from the agents. Each agent is supposed to predict the future prices of the trading stocks in order to maximize its utility function. Each agent has two actions: 1) selling its stocks, 2) buying other agents' stocks for maximizing its expected profit over time.[2]

The stock price model simulated by the author is stochastic, dynamic and dependent by the agents' strategies. In the mathematical model of the stock price, each agent has a different effect (or weight) on the price based on its market share (and market power). Each agent has to maximize its expected profit function with respect to the price constraint. We use the Genetic Algorithm for each agent to optimally select its action (selling or buying stock) in the market and evaluate its fitness value. The author has summarized the entire market model which is necessary to understand the soft computing logic of weights on which the genetic algorithm runs. Each stock has its own weight and each agent has its individual weight for that stock. Hence the stock price model will look like:

$$\lambda_{k+1}^{j} = A_k^j \cdot \lambda_k^j + \sum_{i=1}^{n} B_k^{i,j} \cdot u_k^{i,j} + w_k^j$$

For j=1,…, ns

Where,

$$B_k^{i,j} = \frac{S_k^i \cdot K_k^{i,j}}{\sum_{j=1}^{ns} K_k^{i,j} \cdot \sum_{i=1}^{n} S_k^i}$$

For i=1,…, n and j=1,…, ns

Agent i's expected profit function is:

$$\pi_k^i(u_k^i, u_k^{-i}) = \underset{w_k}{E}\{(\lambda_k - \lambda_{k-1})^T u_k^i\}$$

For i=1,…, n

Where,

$$\lambda_k = \left[\lambda_k^1 \quad . \quad . \quad . \quad \lambda_k^{ns}\right]^T$$

$$u_k^i = \left[u_k^{i,1} \quad . \quad . \quad . \quad u_k^{i,ns}\right]^T$$

Sinsce we have simulated a zero sum its equation would look like:

$$\sum_{i=1}^{n} \pi_k^i (u_k^i, u_k^{-i}) = 0$$

Legend of all the variables used:

In the above equations:

λ_k^j : The price of stock-j at period k.

$A_k^j \in N(1,0.25)$; The weighting factor of stock-j's price from period (k) to (k+1).

$B_k^{i,j}$: Agent-i's weighting factor on stock-j's price at period k.

$u_k^{i,j}$: Agent-i's decision variable for trading (selling or buying) stock-j at period k.

$u_k^{-i,j}$: Agent-i's opponents' decision variables for trading (selling or buying) stock-j at period k.

Note:

$u_k^{i,j} \leq 0$ When agent-i sells stock-j,

$u_k^{i,j} > 0$ When agent-i buys stock-j.

$w_k \in N(0,1)$; Stock-price volatility in the market.

S_k^i : Agent-i's wealth for trading in the stock-market at period k.

$K_k^{i,j}$:Agent-i's holding of stock-j at period k.

n : The number of stock market players,

Usually stock pricing models have been done using stochastic time series models. As the author only intend to run test cases he has modelled the stock model genetically and in a stochastic dynamical equation including the effects of agents acts in the market (as control variables). Each agent starts in the stock-market game with the equal amount. The weighting factors for all stock-prices at the starting period are equal. The weighting factor of an agent on a stock-price depends not only on its market share for that stock, but also on its market wealth at that time. [2]

The authors have modelled the stock-market as a sequential-game in scenario 1 where the rank of each agent in the decision-making process changes from one trading period to another. This makes it a non-symmetric information game. As we can imagine, the first agent in the decision making process has the least information about other agents' acts, but on the other hand it has the advantage of trading first and affecting the stock price before others. In second scenario, they have modelled the stock-market in a non-sequential game where in each trading period; an agent trades with the stock broker. Each agent is chosen by a priority (based on a probability factor) to enter a trading period. The bigger the probability factor, the more likely is the agent to enter in game. [2]

As the game is non-linear and of stochastic nature the author has chosen GA algorithm for solving them. The constraints in the optimization problem using the GA are: 1) Total number of each stock is fixed during a trading period, 2) An agent with zero wealth can only sell stocks, 3) An agent with no stocks and positive wealth can only buy stocks. These constraints are considered in our GA solution method. [2]

In the final study, the author is only considering fundamental quality of stocks described by the basic parameters which include firms' share price rationality, growth, profitability, liquidity, efficiency, and leverage attributes. In general, these fundamental variables can be used to determine the value of a stock, defined by the score assigned by the authors proposed model. These scores are assigns using the hybrid fuzzy-GA algorithm and at the end of the day the stocks with a higher score are picked to create portfolio. [3]

The author has taken a linear approach which uses the fundamental variables to score stocks. Let $X_{i,j,t}$ denote the score of stock i assigned by variable j at time t, where $X_{i,j,}$ depends on the value of variable j, $v_{i,j,t,}$ for stock i at time t. the simple linear formula derived for this is:

$$X_{i,j,t} = p_{i,j,t},$$

Where $p_{i,j,t,}$ belongs to N i.e. the ranking of stock I with respect to variable j at time t. Here we denote a stock sorting indicator Ij for variable j and consider two cases for the stock sorting scheme:

(1) $I_j = 0$: $\rho_{i,j,t} \geq \rho_{k,j,t}$ iff $v_{i,j,t} \geq v_{k,j,t}$ for $i \neq k$.

(2) $I_j = 1$: $\rho_{i,j,t} \geq \rho_{k,j,t}$ iff $v_{i,j,t} \leq v_{k,j,t}$ for $i \neq k$.

Let W_j denote the weight of the j-th variable. Then the total score of stock i at time t, $S_{i,t}$, can be defined as:

$$S_{i,t} = \sum_j W_j X_{i,j,t}.$$

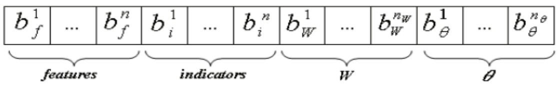
The author has applied the fuzzy logic scoring model for better accuracy and flexibility so that the weight of the outputs of the genetic algorithm can be adjusted proper improve the accuracy of the stock scoring model. The calculated outputs by the GA may be adjusted properly to reflect the actual potential of stock price increase in the future. This adjusted score, denoted by $y_{i,t,}$ is determined by the score calculated. Fuzzy algorithm is the best algorithm in soft computing that shifts weights around to give the best output. [3]

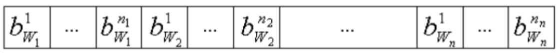
$$y_{i\,t}(W, \theta) = S_{i,t}(W, \theta) * \mu(S_{i,t}(W, \theta)),$$

Y is the new scores is the old score, W are the weights and theta are features chosen for the stock, t is the time, i is the ith stock and u (mew) is the fuzzy membership function. [3]

The fuzzy logic set for stock selection isn't good enough to figure out the best case solution for the scoring mechanism. The evolutionary algorithm are better suited to deal with feature interaction problems. Hence the genetic algorithm is used to solve the scoring mechanism generation by generation. [3]

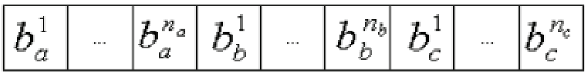
The most important part of genetic algorithm is the chromosome formation. This chromosome is the binary encoded dataset which is considered to be the child in the algorithm. The chromosome formation is the most important part in this algorithm. Here we employ the GA for simultaneous optimization of these tasks. In our overall encoding design, the composition of a chromosome is devised to consist of four portions - the candidate set of features F, the stock sorting indicators I, the weighs Wand the fuzzy model parameters T. In this study, the binary coding scheme is used to represent a chromosome. [3]





The chromosomes are encoded with features, indicators, parameter functions and weights

The encoded fuzzy model parameters



The second most important part of the genetic algorithm is the calculation of the fitness function. After every generation is computed we derive a fitness function and if the fitness function. The fitness function for this genetic algorithm is done by using a simple mathematical formula comprising of the annualized return of a stock.

$$fitness = \sqrt[n]{R_c},$$

This method doesn't follow the game theoretical approach in its strictest sense but the working of the genetic algorithm closely resembles the zero-sum game.

VI. Co-evolutionary Algorithm

The co-evolution algorithm is the newly developed algorithm used to solve the stock selection game because fuzzy or GA algorithms weren't good enough independently. So, the authors have combined both the algorithm to formulate a co-evolutionary Hybrid Fuzzy-GA algorithm. The fuzzy rule-base was evolved using an evolution strategy. Both recombination and mutation are used as variation operators. Each offspring is created by randomly choosing two parents. After the offspring is born a standard I-point crossover, followed by mutation, transforms it into a better generation. Mutation added a Gaussian distributed random variable with zero mean and standard deviation of around 0.2. For the defuzzified output any value below a certain range was automatically set to 0 because a desirability of buying stock with a value that low makes no sense. [1]

The author proposed fuzzy-GA model for stock selection is a multi-stage process, including feature selection and parameter optimization by the GA, stock scoring, score fuzzification, stock ranking and selection, as well as performance evaluation. The flowchart of this hybrid algorithm is shown below. [3]

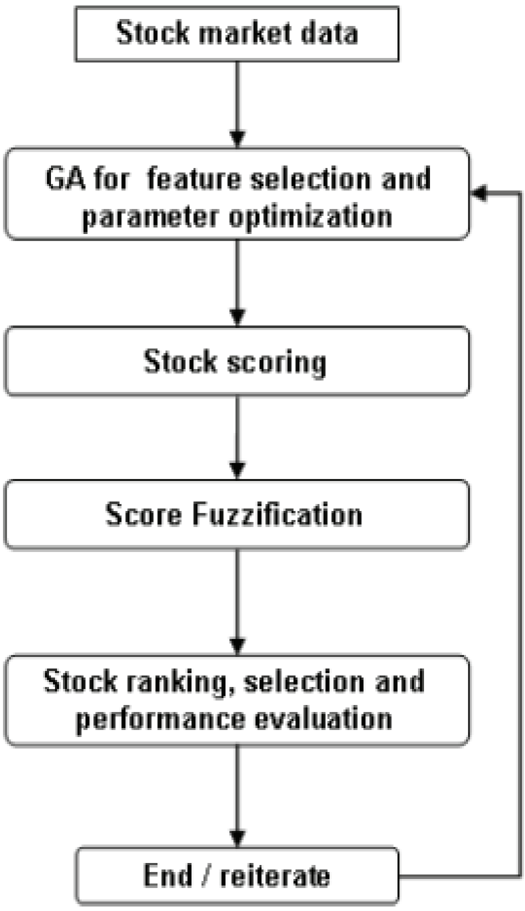


Fig. 4. Hybrid fuzzy-GA Co-Evolutionary Algorithm.

This co-evolutionary algorithm is run for many generations to get a desirable output and to train the genetic and fuzzy neural network. These iterations are necessary because only after so much processing the program will understand to shift the weights around and yield better results.

VII. EXPERIMENTAL RESULTS

In the first reference paper the author has taken a large number of trading periods to train the system to adapt to the market. Since it is a zero sum game the players are all started out with equal amount i.e. $400. The first 1000 days are reserved for training whereas the 600 remaining days are used for validation testing. Each training simulation consists of 150 consecutive trading days partitioned into three overlapping 90 day segments. The second segment used the last 60 days from the first segment plus 30 new days; the third segment used the last 30 days of the first segment plus 60 new days. The overlaps help mitigate market volatility. Each training epoch started with a $400 bank account. The bank balance at the end of the first 90 day training segment was the beginning bank balance of the second 90 day segment. Similarly the third segment beginning balance was the same as the second segment ending balance. (Bank account balances are adjusted as explained in the previous section.) Testing was conducted in the same way, except the 150-day window was chosen from the latter 600 days in the database. The figure below shows the run of the fuzzy output rule-base where the shares were bought if the score was above a certain point.

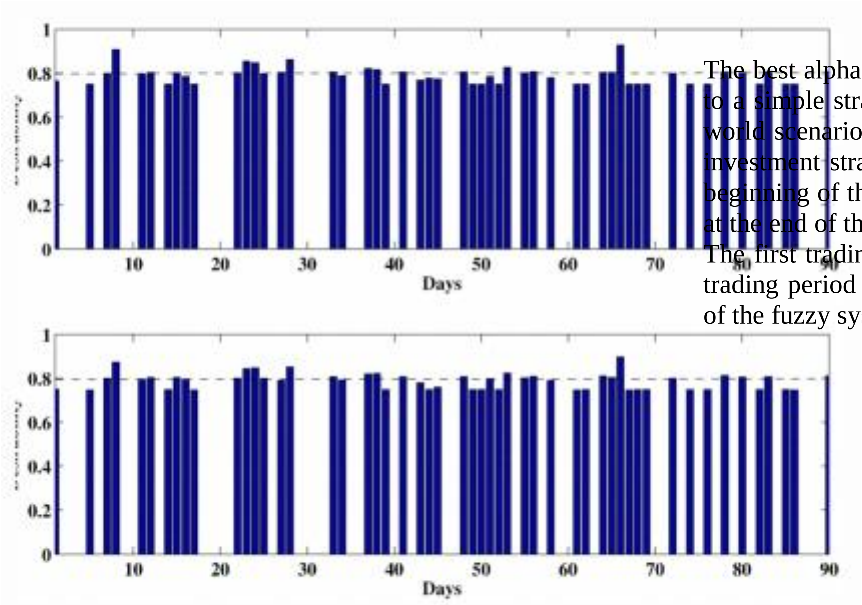


Fig. 5. Typical trading behavior for the best fuzzy trading rule sets over a 90 trading period. Shares were bought whenever the desirability exceeded 0.8.

This data is only the buying and selling of a certain stock. It doesn't exactly tell us the rise and fall of the utility function for an investment firm. The table below shows the bank account balances for the top five and bottom five trading strategies after completing the 150 days of trading using the training database. For both firms the top five strategies had positive returns whereas the bottom five strategies had negative returns. The positive returns are considerably higher than the starting balance of $400. Good performing strategies have a higher pay off by taking money out of the bank accounts of poorer performing strategies. This zero-sum game format matches how investors in the real world choose brokerage firms: firms with good strategies attract more investments and firms with bad strategies keep bleeding money.

	Account Balances from Top 5 Strategies ($)	Account Balances from Bottom 5 Strategies ($)
α Firm	548.4, 498.2, 489.6, 487.8, 479.6	336.1, 335.4, 332.2, 331.1, 326.2
ω Firm	548.9, 548.7, 548.3, 547.6, 547.0	322.2, 321.2, 319.0, 304.5, 300.1

BANK ACCOUNT BALANCES FROM TOP AND BOTTOM TRADING STRATEGIES FROM TWO BROKERAGE FIRMS. RESULTS ARE OBTAINED OVER A 150 TRADING PERIOD WITH AN INITIAL $400 BANK BALANCE.

The best alpha firm fuzzy rule-base strategy is then compared to a simple strategy being followed in the market in the real-world scenario. The test market data is used. In the simple investment strategy fifty shares of stock are purchased at the beginning of the trading day and then sold at the closing price at the end of the trading day. Two trading periods were chosen. The first trading period begins on day 1200 while the second trading period begins on day 1400. The figure for the payoff of the fuzzy system and simple system is given below:

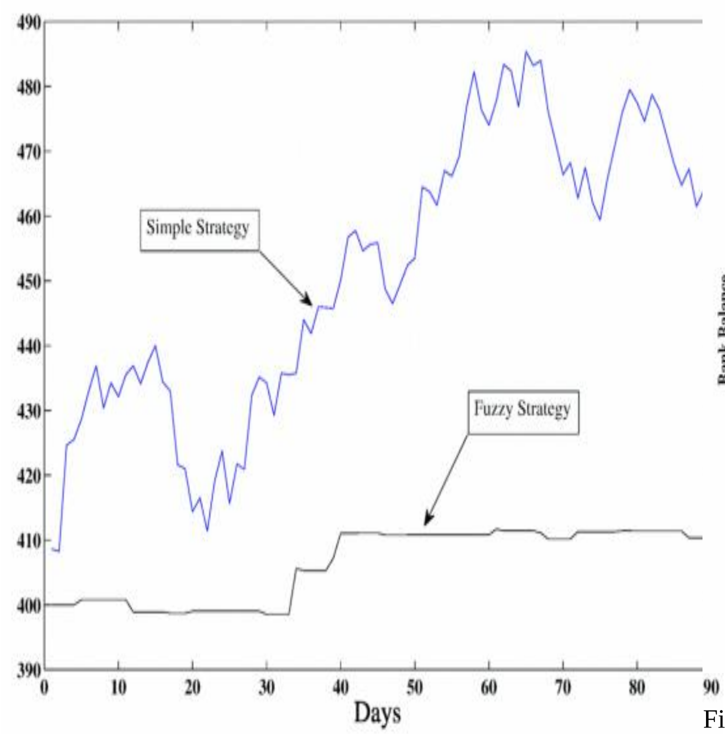


Fig. 6. Bank account balances for the top alpha firm strategy and the simple investment strategy over a 90 day period of test market data starting at the 1200th trading day. Both accounts started with a $400 balance.

The most interesting observations is that the simple strategy is better than the fuzzy strategy. But this is because the market data shows high volatility until the 1200th day and after that it reverses suddenly. Evidently the simple strategy traded the shares every day and quite a good run but after 1400th day we get a chart like Fig 7.

Now evidently the simple strategy most certainly had a gratuitous payoff until the market was unstable but once the market got to be stable the simple strategy failed and the investors lost a huge sum of money.

Whereas in the fuzzy strategy the income may not be as high as anticipated but it most certainly had a modest income irrespective of the state of the market.

Hence the fuzzy strategy yielded a steady 2.5% income even when the market was down.

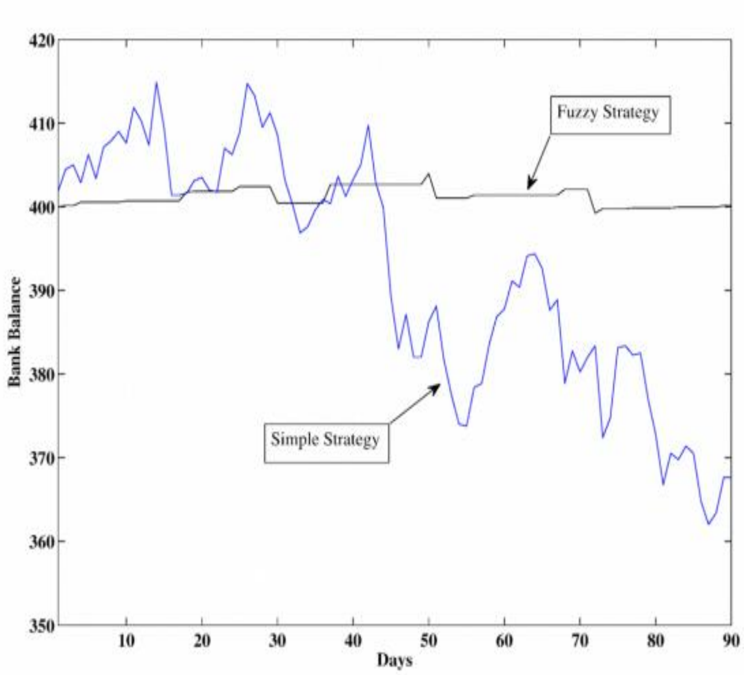


Fig. 7. Bank account balances for the top alpha firm strategy and the simple investment strategy over a 90 day period of test market data starting at the 1400th trading day. Both accounts started with a $400 balance.

In the second paper the authors have taken two case scenarios. In the first scenario there are eight different stocks (with certain initial prices) distributed among four agents. Agents have equal initial wealth of $1,000,000 each. In each trading period, agents randomly change positions in the sequential decision making process. The 4 agents are allowed to trade for 100 training periods. In the graphs below the positive value of the stock means buying and the negative value is selling. The total wealth of each agent after 100 trading periods will tell us which agent had the best outcome in their respective generation. Since, we had a zero-sum game the total market wealth remained fixed ($4,000,000). [2]

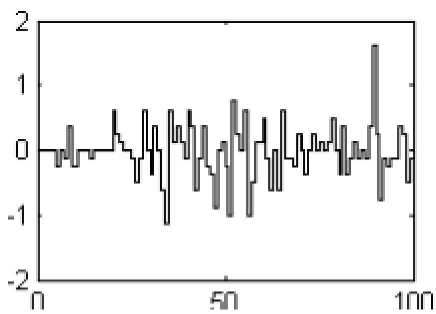


Fig. 7. Optimal trading decision of agent 1 for 100 rounds in scenario 1

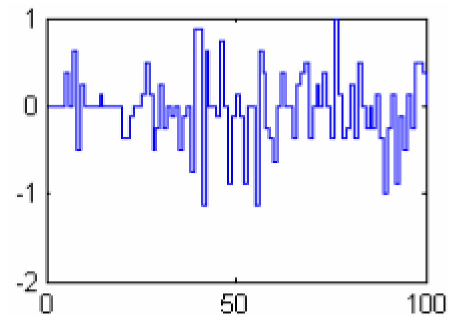


Fig. 8. Optimal trading decision of agent 2 for 100 rounds in scenario 1

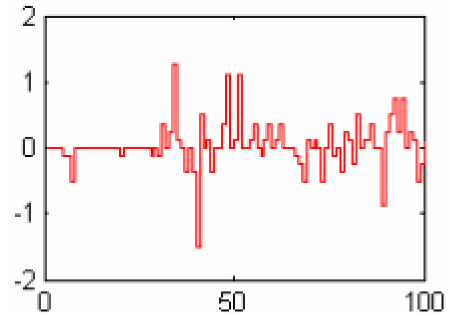


Fig. 9. Optimal trading decision of agent 3 for 100 rounds in scenario 1

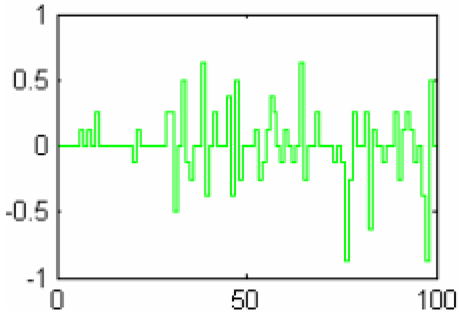


Fig. 9. Optimal trading decision of agent 4 for 100 rounds in scenario 1

The total wealth accumulated after 100 trading days by the 4 agents is listed below in the table.

Final wealth of each market player in scenario 1

Market Players	Total wealth [in USD]
Agent 1	1.48 e+006
Agent 2	1.45 e+006
Agent 3	1.47 e+006
Agent 4	3.61 e+006

It is clearly visible from the table that agent 4 has made the most profit. This is because agent 4 has sold and bought shares in an equilibrium. Whereas agent 1 has sold the most amount of shares and agent 3 has not dealt in many shares at all. Where the simple rule applies you only get as much as you put in the pot.

The second scenario is much more interesting that the first cause the agents are kept on a pre-condition to deal in stocks with. In the second case study there are eight different stocks (with certain initial prices) distributed among three agents and a stock broker. The Agents have equal initial wealth of $1,000,000 each and the stock broker has $3,000,000. In each trading period, one agent trades with the stock broker. The probability factors for agents 1 to 3 were given as zero, 0.70 and 0.30 respectively.

The following graphs show the trading decisions by the 4 agents.

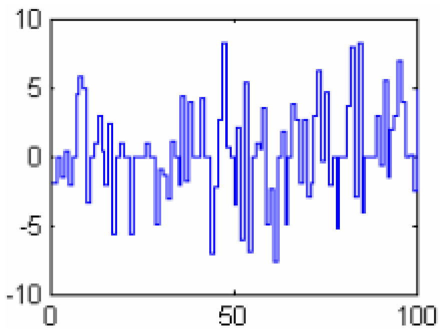


Fig. 10. Optimal trading decision of agent 2 for 100 rounds in scenario 2

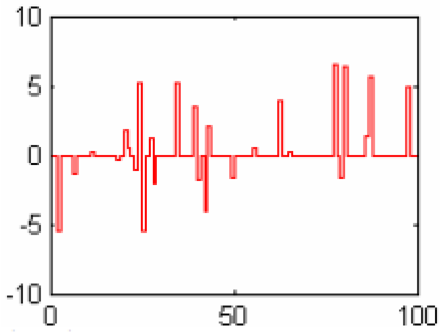


Fig. 11. Optimal trading decision of agent 3 for 100 rounds in scenario 2

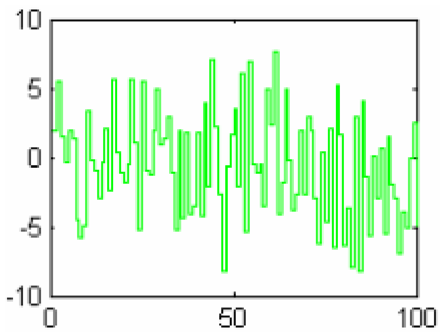


Fig. 12. Optimal trading decision of agent 4 for 100 rounds in scenario 2

Final wealth of each market player in scenario 2	
Market Players	Total wealth [in USD]
Agent 1	1.59e+006
Agent 2	2.09e+006
Agent 3	1.70e+006
Stock broker	4.13e+006

The table shows the wealth of each agent after 100 rounds. Comparing with agent one who did not trade at all, agent two with the highest probability factor made the most profit and agent three that trade more than agent one but less than agent two, make more profit than agent one but not as much as agent two. From our market simulation studies we observed that in highly stochastic and uncertain trading environments, optimal strategies are not always the winning ones. Whereas in such markets, repetition and learning the behaviours of the opponents and dynamic information could become the main factors in making the optimal decision for each market participant. Therefore, combination of game theory and Genetic Algorithm (as a mathematical tool for solving nonlinear complex optimization problems) could help us to solve non-linear repeating market games and discover the optimal trading volumes and stock prices. [2]

In this study, standardization was first applied to the research data every original attribute is scaled into the range of by subtracting the mean, and dividing the result by the standard deviation. This treatment is to ensure that all the attributes lie in the same parameter range, in order to prevent attributes with large ranges from overwhelming others and prediction errors may be reduced. To examine our proposed model, stock data of all the years is first used for optimization by the GA. Stocks are ranked based on the scores obtained. Top stocks are selected and Figure 6. Best-so-far curves by the GA the average yearly return of these selected stocks are calculated. The data is generalized over 50 runs attained by the GA over 50 generations. The averaged best-so-far performance curve is calculated by averaging the best-so-far obtained at each generation for all 50 runs, where the vertical bars overlaying.

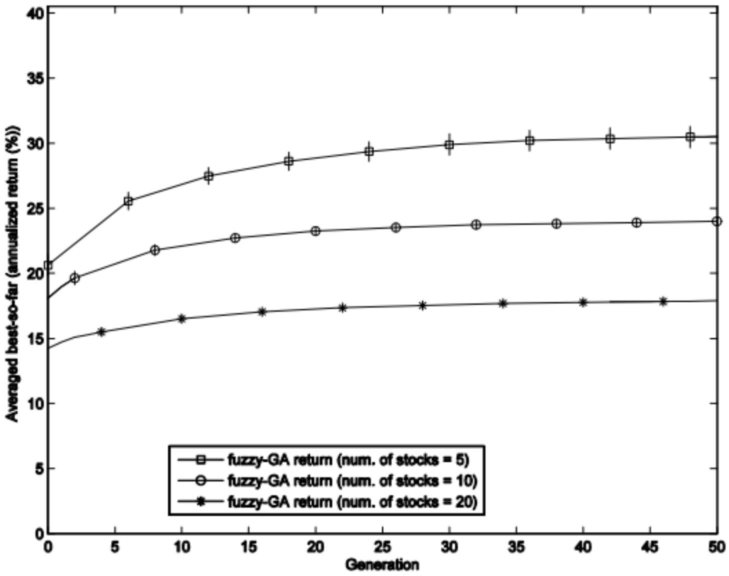


Fig. 12. Best so-far curves

The table for the attributes taken into consideration for the co-evolution hybrid fuzzy-GA logic is as follows. These list of attributes help us in calculating the fitness function and nature of the trend of the stock.

Attributes used in the stock selection model

Attribute	Ratios	Description
Share price rationality	PE ratio	Price-to-earnings ratio = share price / earnings per share
	PB ratio	Price-to-book ratio = share price / book value per share
	PS Ratio	Price-to-sales ratio = share price / sales per share
Profitability	ROE	Return on equity (after tax) = net income after tax / shareholders' equity
	ROA	Return on asset (after tax) = net income after tax / total assets
	OPM	Operating profit margin = operating income / net sales
	NPM	Net profit margin = net income after tax / net sales
Leverage	DE ratio	Debt-to-equity ratio = total liabilities / shareholders' equity
Liquidity	CR	Current ratio = current assets / current liabilities
	QR	Quick ratio = quick assets / current liabilities
Efficiency	ITR	Inventory turnover rate = cost of goods sold / average inventory
	RTR	Receivables turnover rate = net credit sales / average accounts receivable
Growth	OIG	Operating income growth rate = (operating income at the current year – operating income at the previous year) / operating income at the previous year
	NIG	Net income growth rate = (net income after tax at the current year – net income after tax at the previous year) / net income after tax at the previous year

To display the effectiveness if there algorithm the authors have run the test on the real-world data alongside a benchmark which is a simple strategy used by investment firms in the present day. Here is an illustration of the cumulative benchmark return (the product of the average yearly returns of the 50 stocks in the investment universe) and the cumulative returns of longing a number of top-ranked stocks recommended by the fuzzy-GA model. This figure shows that the portfolios of maintaining 5, 10 and 20 stocks outperform the benchmark at the end of year 2009. As a result, the optimization on the stock selection model by the GA is advantageous to stock selection.

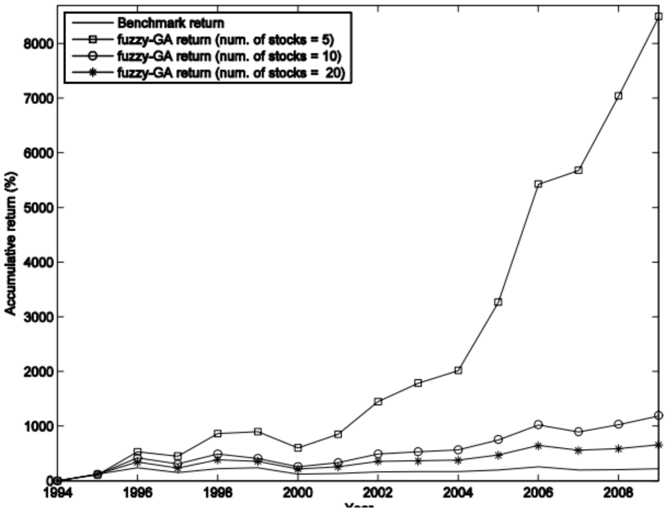


Fig. 13. Cumulative returns of benchmark v.s. longing top-ranked stocks by the fuzzy-GA stock selection model

To further show the efficiency of the algorithm the statistical data is also being presented for the case of 5 stocks. This table comprises of the Statistics of the benchmark and fuzzy-GA stock selection models for 5 stocks.

Training period	Annualized benchmark return	Mean of annualized model return	Standard Deviation	Standard error	Testing period	Annualized benchmark return	Mean of annualized model return	Standard Deviation	Standard error
1995	23.1487	61.4170	2.61412	0.36969	1996-2009	4.25672	1.1731	5.34558	0.75598
1995-1996	53.9684	162.6117	10.33728	1.46191	1997-2009	-0.54716	-0.5062	4.02765	0.56959
1995-1997	15.3332	76.9770	3.14302	0.44449	1998-2009	3.07863	5.2559	5.66329	0.80091
1995-1998	22.4658	87.1291	5.02817	0.71109	1999-2009	-0.17093	0.1383	5.11039	0.72271
1995-1999	19.1951	69.1519	5.39052	0.76233	2000-2009	-0.85761	0.9131	5.79058	0.81891
1995-2000	3.7129	43.7078	3.15297	0.44589	2001-2009	6.57475	10.2523	7.24027	1.02393
1995-2001	4.0824	40.2833	2.63405	0.37251	2002-2009	6.60575	13.3479	4.55598	0.64431
1995-2002	6.5905	41.9213	2.32244	0.32844	2003-2009	4.09937	10.1117	5.37307	0.75986
1995-2003	6.1517	39.6470	2.09784	0.29668	2004-2009	4.33343	7.6255	7.19403	1.01739
1995-2004	5.5582	36.0228	1.25193	0.17705	2005-2009	5.14611	11.0043	8.36337	1.18275
1995-2005	6.5087	35.5869	1.82544	0.25815	2006-2009	2.48545	9.02378	7.69611	1.08839
1995-2006	8.124	36.9053	1.67149	0.23638	2007-2009	-4.73667	-1.64134	9.27202	1.31126
1995-2007	5.5105	33.5921	1.64090	0.23205	2008-2009	4.83837	5.25787	9.33182	1.31972
1995-2008	5.3590	31.5289	1.71741	0.24287	2009	6.28742	15.3057	12.88159	1.82173

VIII. CONCLUSION

In all the papers the authors have modelled the stock market as a stochastic non-cooperative repetitive game with. The problem with the stock game i.ee is strategy formulation is done by computational intelligence. Everything in basic fundamental sense of stocks is taken into variables in the market (selling or buying stocks) at different trading periods and a firm decision of buying or selling has been made. Best part of the papers is that they have applied the Genetic Algorithm for solving our stochastic stock-market game and the fuzzy logic is used for shifting weights for stock scoring mechanism. Our market simulation results illustrate that the agents' information and optimization algorithms (optimal selection of the GA algorithm parameters) could have considerable effects on their profit maximization. Also an agent is more successful by making more decision in the trading rounds with the broker. Finally simulations studies show that the genetic algorithm (GA) could be a good candidate for analysing financial markets due to its strength for optimizing non-quadratic profit (or fitness) functions and non-linear price behaviour in the market. In our future studies we plan to use more complicated models for the stock-price (including price elasticity) and players' objective functions in the stock market and test the accuracy and robustness of the GA algorithm for solving such games.

These algorithms are not only just applied but a new one is formed by combing fuzzy logic and genetic called the Co-evolutionary hybrid fuzzy-GA methodology for stock selection. Based on the designed scoring mechanism for a set of stocks, top-ranked stocks can be selected as components in a portfolio. In the meantime, the GA was employed for optimization of model parameters and feature selection for the model. This algorithm has even shown better results than the strategies being used by investment firms these days. Therefore, this hybrid model to advance the research in computational finance and provide a promising solution to stock selection in practice.

The research on this subject that I have done has truly been invigorating and I have learnt that something as unprecedented and stochastic can be solved using a computer. If this were ever to be implemented then it would a huge step to the development of AI. Also the stock market has been devised into non-cooperative multi-agent game and it has been computerized using soft-computing logic which is truly amazing. The best part is if it can make money at a steady pace.

IX. FUTURE WORK

Most of the work done by the authors and the people before them is very preliminary and in its adolescence. It may work on the training data but it mostly certainly hasn't been tested on real world data. Also all the algorithms require a lot of learning time until which no profit is reaped. These algorithms have been fed in a lot of refined data which has been made to fit its parameters and in a lot of places they require human input which means they are not completely automated.

In most of the test cases all the brokerage firms used the same membership function which is not really possible because in the real-world different firms have different strategies and thus different membership function.

Also the algorithms were only designed to earn better yields which is to buy profitable stock but sometime one must unload some stock that might render them losses.

In the future, a plausible research direction is to employ more advanced fuzzy and scoring models to investigate how performance of portfolios can be further improved. In addition, in our current model, we consider the first several years as the training set and the next several years as the test set. This may not be sufficient to generate a feasible model. Therefore, in the future work, a model that would be capable of generating more time-dependent patterns to account for the impact of the more recent past of the stocks.

ACKNOWLEDGMENT

I would like to thank Professor Quanyan Zhu for his support and assistance in this study and encouraging more and to take up a project like this and learn a lot in the process.

REFERENCES

[1] Garrison W. Greenwood and Richard Tymerski, *A Game-Theoretical Approach for Designing Market Trading,* IEEE

[2] Armin Zeinali and, Ashkan Rahimi-Kian, *A Game-Theoretic Based Dynamic Stock Market Modeling* ,IEEE

[3] Chien-Feng Huang, Chih-Hsiang Chang ,Bao Rong Chang and Dun-Wei Cheng, *A Study of a Hybrid Evolutionary Fuzzy Model for Stock Selection,* National University of Kaohsiung ,2011 IEEE International Conference on Fuzzy Systems