Genetic Algorithm: An Application to Technical Trading System Design

V. Kapoor
Institute of Engineering & Technology,
Devi Ahilya University, Indore

S. Dey
Indian Institute of Management
Indore

A.P. Khurana
School of Computer Science
Devi Ahilya University, Indore

ABSTRACT
Recent studies have shown that in the context of financial markets, technical analysis is a very useful tool for predicting trends. Moving Average rules are usually used to make "buy" or "sell" decisions on a daily basis. Due their ability to cover large search spaces with relatively low computational effort, Genetic Algorithms (GA) could be effective in optimization of technical trading systems. This paper studies the problem: how can GA be used to improve the performance of a particular trading rule by optimizing its parameters, and how changes in the design of the GA itself can affect the solution quality obtained in context of technical trading system. In our study, we have concentrated on exploiting the power of genetic algorithms to adjust technical trading rules parameters in background of financial markets. The results of experiments based on real time-series data demonstrate that the optimized rule obtained using the GA can increase the profit generated significantly as compare to traditional moving average lengths trading rules taken from financial literature.

Keywords: Genetic Algorithms (GA's), Population size, Trading system, Technical rule.

1. INTRODUCTION
Investors in the stock market are concerned about maximizing rate of return or profit. In recent years global stock markets were bullish in trend, and there was significant increase in the number of researches focusing on stock market investing. Due to huge amount of data available, investors faces difficulty in making decisions, concerning how to earn higher profits from stock trading. Many experts in financial theory have failed in their attempts to predict the trend in market completely, due to the noisy and fluctuating trend of prices. The Efficient Market Hypothesis (EMH) is one of the most important basics of modern finance theory which states that it is impossible to beat the market.

Technical analysis is a popular method used in stock trading. There is a lot of literature devoted to technical analysis rules that are supposed to be able to identify trends (bullish or bearish) or reversals in trajectories of prices [4, 8, 9, 10]. Technical analysis assumes that future trends can be recognized as it is function of past prices. For last few years there has been concentrated discussion among practitioners and academicians about utility of technical trading rules and the Efficient Market Hypothesis (EMH). Early attempts to use technical indicators were based on simple filters [2]. While analyzing literature on technical analysis available one can feel uneasy at times, due to the availability of virtually infinite number of technical trading rules the multitude of ways in which they can be applied.

More recently moving averages have been used by Brock et.al [2], their work shows that profitable results could be obtained using this method. A simple moving average system, commonly used in trading simulators, and has two set of parameters (the lengths of two moving averages). As moving average lengths can cover various time frames (from 1 day to 500 days etc.), the use of the correct parameters (i.e. the lengths) in constructing trading rules is immensely important. Also, there exist a large number of various parameters combinations for moving average rules making it impossible to test all of them manually as a part of the decision making process. Therefore, it is imperative to develop automated methods for optimizing trading systems.

Optimization is a search process by which the 'best' solution can be discovered. It is simply a method or algorithm that allows us to find the best or close to best possible solution for a given problem. In case of trading system or rule optimization this can be done to find a particular set of rule parameters. The best existing solution or fitness for a given problem can be discovered in a number of ways. Some simple problems are solved by trial-and-error, often guided by human insight since human brain is one of the most powerful heuristic optimization systems on earth. In some cases, analytic optimization such as developing multiple regression models [1, 3, 5, 6, 7] or calculus based procedures are used. For more complex systems complicated methods or algorithms are needed. In real terms it is an algorithm implemented with programming code in some computer language in case of use of computers. Simulating the process of evolution (as genetic algorithms do) is a very novel way to discover or find high quality solutions to complex problems.

GAs are heuristic algorithms based on 'survival of the fittest' principle, and do not guarantee a globally optimal solution, only a close to optimal one. They have been formally shown to be a remarkably efficient approach for optimizing non-linear functions [4, 8, and 9]. The fitness function used in GAs is a block of programming code that reflects the attractiveness of a particular solution. In case of our problem the fitness is interpreted as net profit for intense drawdown.

The aim of the work described in this paper is to investigate how genetic algorithms, a class of algorithms in evolutionary computation can be used to improve the performance of a particular trading rule, and how changes in the design of the GA itself can affect the solution quality obtained in context of technical trading system. Here our main aim is to discover an optimal set of trading rule parameters evaluated on a real financial time series (i.e. historical data). It is not our purpose to provide theoretical or experimental justification of technical analysis.

This paper is structured in the following way: Genetic Algorithms and related work are presented in section 2, Data set and methodology is then described in section 3, Section 4 explains the proposed GA based system, Section 5 is devoted to the performance study of this approach. Conclusions and some extensions proposed follows Section 6.

2. LITERATURE SURVEY

Genetic Algorithms are search and global optimization procedure based on the principles of natural biological evolution. GA is an attempt to combine computer science and natural evolution. It is an attempt to imitate power of natural evolution in computer program. Darwin theory of natural selection is inspiration for GA.

GA is stochastic in nature, such that they take advantage of random chance in their operation. GA begins with random chosen population of candidates who are evaluated accordingly to fitness function defined and then newer solutions are evolved using genetic operators such as selection, crossover and mutation see fig 1.

```
Genetic Algorithm ()
{
        Generate initial selection operator
        Evaluate the fitness value
        While ( do not satisfy a stop condition)
        {
            Execute the selection operator
            Execute the crossover operator
            Execute the mutation operator
            Evaluate the fitness value
            }
}
```

Figure 1: Genetic Algorithm

Beginning with random generation, GA goal is to improve the fitness of solutions as the generation passes by. The stopping condition is specified as some fixed number of generations reached or achievement of satisfactory fitness level. The reasoning behind crossover is that if two parents are represented by high level of fitness, crossing them will lead to better offspring (solutions). As search space is large, some amount of randomness is maintained through generations by mutation operator applied infrequently, so as population does not looses its genetic diversity. Main aim of GA is to discover almost optimal experts or solutions. GA is characterized by speed. It is many times faster than brute force optimization method especially when faced with combinatorial explosion.

Application of GA to stock trading has already been evaluated by a number of researchers. GA on financial applications have shown promising results. Bauer [18] used GA to generate trading rules which are Boolean expression. To my awareness the first paper linking GA's to investments was from Bauer and Liepins [19]. Baur [19] in his book "Genetic Algorithms and Investment strategies" offered realistic guidance concerning: How GA's might be used to develop striking trading strategies based on fundamental information? These techniques can be easily extended to include other type's information such as technical trading rules as well as past prices. According to Allen and Karjalainen [20], genetic algorithm is an appropriate method to discover technical trading rules. Some other interested studies is done by Mahfoud and Mani [21] that present a new genetic algorithm based system and applied it to the task of predicting future performances of individual stocks and genetic programming to foreign exchange forecasting and reported some success. Goldberg [15] and De Jong [22] suggest that a set control parameters (Crossover 0.6, mutation 0.0333, population size 300 works well across many problems. Bauer [18] performed a series of simulations on financial optimization problems and established the vigor of Goldberg suggestions. Ramon Lawrence [23] studied methods of using GA's to train a neural network trading system. Korczak et.al [24] used GA to search a set of trading rules which gives buying and selling signals on individual stocks. Jin Li et.al [25] used genetic programming to improve technical analysis predictions. Laura Nu´n˜ez-Letamendia [26] and many others [27] applied GA to optimized the parameters of a trading rule. According to Laura Nu´n˜ez-Letamendia [26] GA's work better in high crossover and low mutation probability and a moderate population size. See also [16, 17].

3. METHODOLOGY

Though we can construct an endless number of trading rules by combination of large number of technical indicators and arithmetic operators, some of these rules are popular, and are extensively used by practitioners and attracted the attention of academics. One of these well known rules is "Crossing of Moving Averages". Average is defined as the average price of last n days. The word "Moving" means that set of prices being averaged is continuously moving through time. Moving Average is a lagging indicator which is used to flattened the unpredictable or raucous data in order to get true trend of prices. Trading rule called "Crossing of moving averages" is based on the signals generated by crossing of MA's of different lengths. A moving average for n days is given by:

$$\frac{1}{n}\sum_{i=0}^{n} P_{t-i} \quad \text{.............eq. 1}$$

Where P_{t-i} is the closing price for the day t-i and n denotes the number of past days taken to built moving average (i.e length of MA). Buy and sell signals are generated in the following:

- A buy signal or long position (according to the market talk) is taken when shorter MA is greater than longer MA.

$$\frac{1}{\theta_1}\sum_{i=0}^{\theta_1} P_{t-i} > \frac{1}{\theta_2}\sum_{i=0}^{\theta_2} P_{t-i} \quad \text{............eq. 2}$$

Where $\theta_1 < \theta_2$

- A sell signal or short position (according to the market talk) is taken when shorter MA is less than longer MA.

$$\frac{1}{\theta_1}\sum_{i=0}^{\theta_1} P_{t-i} \leq \frac{1}{\theta_2}\sum_{i=0}^{\theta_2} P_{t-i} \quad \text{................eq. 3}$$

Where $\theta_1 < \theta_2$

The parameters i.e lengths of two moving averages (θ_1 and θ_2) are chosen heuristically or instinctively by the trader. There can be many possible combinations if we confined number of days to built MA to 500, i.e there will be 500 × 500 = 2, 50,000 number of parameters settings for the rule based on crossing of moving averages.

Thus in this way search space increases exponentially i.e the number of parameters to be tuned with the inclusion of technical indicators in the trading system, as every indicator consist of at least one parameter. In case of trading system we have to find combination of parameters that gives best or optimal performance or profit.

In case of crossing of MA based trading system, our main aim is to find parameters i.e lengths of two MA (θ_1 and θ_2) of which trading signals (buy or sell) gives us best return. Thus optimizing our trading system means maximizing our fitness function such as profit which is given by

$$TR_f = \prod_{i=1}^{f} (1 + DR_i) \quad \text{..........................eq. 4}$$

$$DR_i = (P_i - P_{i-1}) \times \delta \quad \text{......................eq. 5}$$

TR_f is the total return for the sample period. DR_i is the daily return for the day *i*, P_i denotes the stock price for the day *i*. δ is the dummy variable which generates value *1* for buy (long) signals and *-1* for sale (short) signals. Trading system takes sell and buy positions, but no out of-market-positions. Transaction costs are not included as our main aim in this paper is to find the possibility of solving an optimization problem by using genetic algorithm with respect to change in population size keeping number of function evaluations constant and to prove robustness of Genetic Algorithm.

4. PROPOSED GENETIC ALGORITHM BASED SYSTEM USING C++ AND VISUAL BASIC:

Genetic Algorithm was developed by Holland is a search and optimization procedure based on the principles of natural biological evolution. GA's differ with other optimization procedures in few ways. GA's work with a coding of parameters not with the parameters individually. Thus GA's can handle binary variables. GA's search from population of points, not a single point. So they can offer globally optimal solutions. Finally GA's works on fitness function not on any other secondary knowledge. Thus GA's are helpful in finding global solutions for non-continuous and non-differentiable functions that actually exist in practical optimization problems.

The system architecture in our research is divided into two parts. Genetic Algorithm module in C++ and fitness function module in visual basic. Our data on which experiment is performed is stored in Microsoft excel.

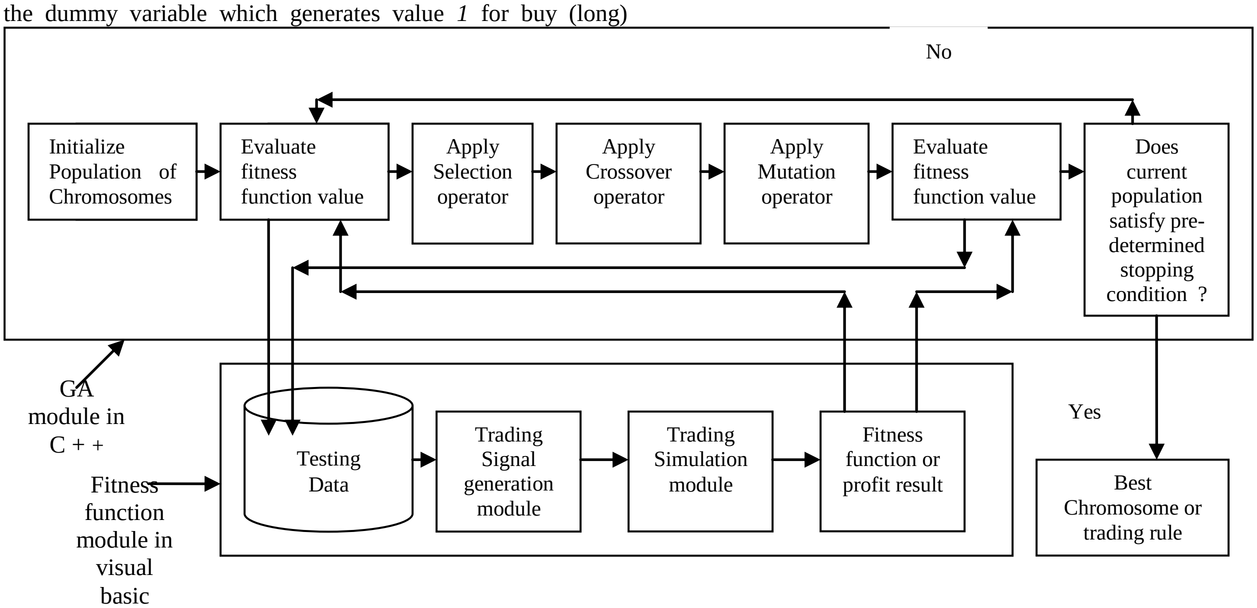

Figure 2: Simulation Process

Each parameter in Genetic Algorithm is encoded as binary string and concatenated to form a chromosome. Shorter moving average goes from 1 session to 256 session i.e string length of 8 bits ex. 01100101], longer moving average goes from 1 session to 512 session i.e string length of 9 bits [ex. 110010101] Thus total search space is of 17 bits [ex. 01100101; 110010101] i.e 2^{17}. In our study we have limited computational resources employed by choosing total number of function evaluations to 2500. We run each experiment with 20, 50, 70 and 100 population sizes, keeping number of function evaluations constant.

It begins with randomly generated population. These solutions are tested accordingly to fitness function given in equation 4 and

5. In order to get better solution for next generation each chromosome exchanges information by using crossover operator imitated from natural genetics to get better solution. Mutation operator is used to add random diversity in the solution [16]. The fitness function is used to measure the solution quality of each chromosome in the population.

We have used "Roulette Wheel" selection method which selects individual probabilistically based on their performance. In case of "Roulette Wheel" selection sum of fitness function of individual chromosomes are calculated. Individuals are then mapped, one to one in continuous intervals in the range [o, sum]. To select an individual a random number is generated from 0 to sum and the individual whose spans the random number is selected. This process is repeated until desired number of individuals are selected. After this individual candidates are allowed to participate in crossover and mutation to produce next generation. In our problem we have kept crossover and mutation probabilities 90% and 1% i.e 0.90 and 0.01.

5. EMPIRICAL RESULTS

In this section, we apply our methodology to "State Bank of India" stock data taken from "National Stock Exchange" of India. The data analyzed consist of 1136 observations on daily closing prices of stock for the period 12/08/2001 to 29/12/2006. The optimization period is defined between 12/08/2001 to 29/12/2006.

Table 1. Population size effect

Population Size	20	50	70	100
θ_1/θ_2	235/414	4//40	12/173	9/212
Max. Profit	Rs. 834.35/-	Rs.932.84/-	Rs. 936/-	Rs. 947/-
Avg. Profit	Rs. 709.46/-	Rs. 745.9/-	Rs. 732.14/-	Rs. 672.02/-
Max. Return	116.60%	141.27%	160.39%	150.38%
Std. dev of Profits	83.029836	98.232199	99.66883	109.248246
Max. profit./St. dev	10.048797	9.4962752	9.3911005	8.66833139

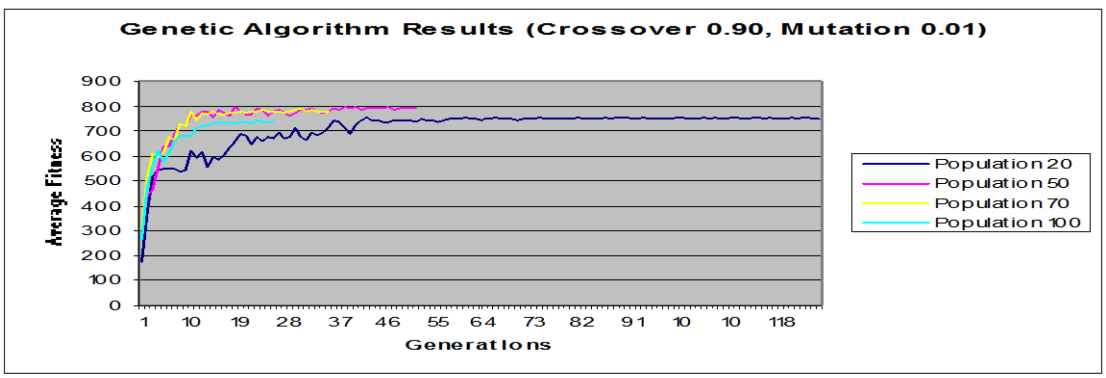

Figure 3: Average fitness function performance for GA applied for different population sizes

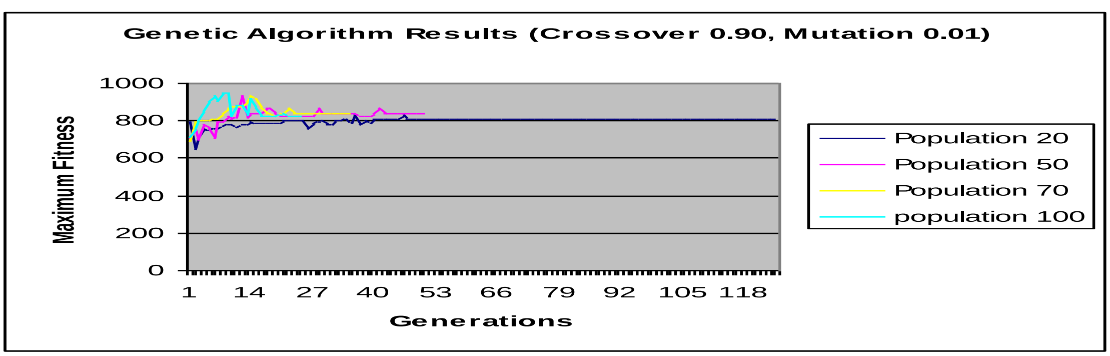

Figure 4: Maximum fitness function performance for GA applied for different population sizes

Our optimization problem is to determine optimal Moving Averages lengths (θ_1 and θ_2) that generates maximize profits. More particularly we are interested to measure the effect of changes in population size (20, 50, 70, 100) in the performance of GA based optimization method keeping number of function evaluations constant to 2500. Our earlier research [16, 17] expresses the concern about lack of study in the particular subject.

Table 1 provided GA optimization results for different population sizes. In order to measure the effect of change in population size in getting best solution we have examine a series of different statistics i.e population size of 20, 50,70,100 with a generation of 125, 50, 35, 25 keeping number of function evaluations constant to 2500. The first row of table shows best parameters i.e MA's Lengths (θ_1 and θ_2) of two moving averages for experiment performed on different population sizes. The second, third and fourth row shows maximum profit or fitness, average profit or fitness and standard deviation of solutions obtained for different population sizes. We have also calculated an efficiency index by dividing maximum fitness by standard deviation of all the solution for different population sizes as shown in the table 1.

By looking at table 1 we can say that as long as population size is increased the best possible solution obtained is higher. In case of average fitness, it increases after population size of 20 and then onwards after population size of 50 it decreases significantly showing that the solution series become noisy in nature and there more diversity in the solution obtained as population size is increased. Best efficiency index is obtained by the population size of 20. It is also seen that lower population size leads to lower performance i.e lower maximum fitness obtained. Standard deviation calculated is the measure of diversity or dispersion in the population. It is seen from table 1 and from figure 3 and 4, that for large population sizes the diversity in the population is large and the best solution obtained is close to the optimal value. Dispersion value increases as the population size is increased which means that volatility in the solution increases with population size. From figure 3 and 4, it is observed that that average fitness and maximum fitness value becomes stagnant after few generations for low population size 20. This is due to the fact that there is less diversity in the population due to this exploration power of genetic algorithm becomes limited thus yielding lesser maximum fitness. From table 1 and figure 3 and 4 it is observed that as population size increases solution quality (Maximum fitness) also increases. Thus in case of large population there is a larger probability of having individuals in population having fitness value or close to the optimal value.

It is in our interest to know the effect of change in population size with respect to number of generations keeping number of function evaluations constant, until a proper balance is reached. This will help us to limit the intensive use of computational resources used by GA, thus reducing the run time and allowing searching larger spaces in a lesser time and computational resource. From our results (see fig. 3 and 4) it is seen for less population size and more number of generations, it leads to convergence at local optimum. In contrast if we increase population size and decrease number of generations, then there will be difficulty in population adapting to the environment. Thus a proper balance between population size and number of generations is needed.

We have seen from figure 4 that in case of population size of 100 we achieved maximum fitness or close to optimal solution at an early generation. However it is seen from table 1 and figure 3 and 4 that changing population size from 50 to 70 does not affect the results obtained significantly. From figure 3 and 4 it is observed that maximum and average fitness follows a positive trend as generations passes by and it becomes stable after few generations for all population sizes. It is seen (Figure 3 and 4) average and maximum fitness improves over time.

Relating to the volatility in the solution standard deviation increases from 83.02 to 109.24 as population size increases from 20 to 100. Due to this maximum fitness also increases from Rs. 834.35/- to Rs 947/- and overall rate of rate of return also increases from 1.16 to 1.50. This is due to the fact as diversity increases with population size and so genetic algorithm ability to find global optimal solution.

It is seen that financial investor or chartists generally test short term, medium term and long term moving averages on past data before taking any decision. Three moving averages lengths are derived from rules in finance literature such as Brock [2]. It is seen that moving averages lengths such as (10,30), (30,60), 50,150) are popular as short term, medium term and long term moving averages lengths in the financial community. Rules generated by GA were tested against these three moving averages lengths. Results are shown in table 2. Among these three moving averages lengths MA (30,60) performs best. It achieve a maximum return of 90.02% and a maximum profit of Rs 535/- . GA based system achieve maximum return of 160.39% and a maximum profit of Rs 947/-, which is better than the MA (30,60) based rule. In fact the poorer GA with a population size of 20 a maximum profit of RS 834.35/- and a maximum return of 116.60% is achieved which is better than these short, medium and long term moving averages based rules.

Table 2. Performance Comparisons between GA generated results for various population sizes and three popular moving averages lengths on the test data

Genetic Algorithm (GA) Results	Max. Profit	Max. Return
GA (Population Size 20)	Rs 834.35/-	116.60%
GA (Population Size 50)	Rs 932.84/-	141.27%
GA (Population Size 70)	Rs 936/-	**160.39%**
GA (Population Size 100)	**Rs 947/-**	150.38%
Highest	Rs 947/-	160.39%
Lowest	Rs 834.35/-	116.60%
Mean	Rs 912.54	142.16%
Std. Dev	52.18	18.74
3 Technical Moving Average (MA) Rules	(the best result in each column is highlighted)	
MA (10,30)	Rs 443.70/-	70.04%
MA (30,60)	**Rs 535/-**	**90.02%**
MA (50,150)	Rs 326/-	78.24%

In all we have attained success in applying GA to achieve optimizing parameters of technical trading systems. Overall high rate of return and profit were obtained (Maximum 160.398%). Superiority of GA is confirmed in terms of high rate of overall return for the test set, illustrating the power of evolutionary algorithms and artificial intelligence in financial engineering.

Examining the above results and experimental data obtained from large number of tests performed, we failed to falsify the theory of Genetic Algorithm strength in finding global optimal solution regarding to Genetic Algorithm control parameters i.e population size and number of generations. To a great extent we are able to beat efficient market hypothesis (EMH) which states that any public information is reflected in the stock price and it is impossible to beat the market.

6. CONCLUSIONS AND FUTURE WORK

Since technical analysis is largely used as a tool in stock trading, it is rarely focused on the issue of parameters optimization. Our main objective in this paper is to demonstrate how new advances in computer engineering and soft computing can be use to improve optimization of technical rules. This system is particularly applied to predict performance of individual stock i.e State bank of India, data taken from National Stock Exchange of India which shows that there is some expectedness in historic data alone.

Starting with moving averages concept which is a versatile, simple and most popular technical indicator used in stock market analysis. We introduce ideas of representing investment strategies as rules, when to buy and when to sell which is put together into conditional statements involving difference of moving averages. Further length of moving averages are encoded as binary strings or chromosomes. By applying GA's operators such as roulette wheel, crossover, mutation, we succeed in using Genetic Algorithm to solve optimization of parameters of technical trading rule with higher overall profit and return obtained.

Our experimental results shows that GA's helps in finding global optimal solutions. We also found that solution quality (maximum fitness) increases as population size increases. Within the impound of limited data set experiments with all population sizes, shows similar results i.e high overall rate of return. It is observed from table 2 there is an increase in maximum return and maximum profit by using optimal moving average lengths obtained from various GAs as compared to the popular moving average lengths obtained from financial literature. Analyzing the experimental data from a large number of tests carried out, our conclusion is that proposition of GA robustness remains reasonable for tuning of parameters of technical trading system. We carried a large number of experiments, but they remain limited as number of test we can perform with respect to population size and number of generations is infinite. So the conclusions are cautious. We are able to beat EMH to a large extent and show that technical analysis has a certain value. Finally it would be motivating for advance research to test a series of different complex trading systems and see how GA performs on other trading systems. This could be our topic of future research.

7. REFERENCES

[1] V. Kapoor, S. Dey, A. P. Khurana"Modeling the Influence of World Stock Markets on Indian NSE Index." Published in the proceedings of International Conference on Modeling and Simulation (MS '09). Organized by College of Engineering Trivandrum and AMSE. Thrivananthapuram, 1st-3rd December 2009.

[2] Brock, W., Lakonishok, J., LeBaron, B., "Simple technical rules and stochastic properties of stock returns." Journal of Finance XLVII (5), 1731-1764. 1992.

[3] Xiaoqing Weng, Junyi Shen, "Detecting outlier samples in multivariate time series dataset." Knowledge-Based Systems. Elsevier. 2008.

[4] Jarl Kallberg, Paolo Pasquariello, "Time-series and cross-sectional excess comovement in stock indexes." Journal of Empirical Finance. Elsevier. 2007.

[5] Chang-Jin Kim, Jeremy Pigerc, Richard Startz, "Estimation of Markov regime-switching regression models with endogenous switching." Journal of Econometrics. Elsevier. 2007.

[6] Nicolaas Groenewold, Sam Hak Kan Tang, Yanrui Wu , "The profitability of regression-based trading rules for the Shanghai stock market." Journal of Empirical Finance. Elsevier. 2007.

[7] Andréas Heinen, Erick Rengifo, "Multivariate autoregressive modeling of time series count data using copulas." Journal of Empirical Finance. Elsevier.

[8] P. Manchandaa, J. Kumara, A.H. Siddiqi, "Mathematical methods for modeling price fluctuations of financial times series." Journal of the Franklin Institute. Elsevier. 2006.

[9] Michael D. McKenzie, Suk-Joong Kim, "Evidence of an asymmetry in the relationship between volatility and autocorrelation." International Review of Financial Analysis. Elsevier. 2005.

[10] R.H. Loschi, P.L. Iglesias, R.B. Arellano-Valle, F.R.B. Cruz, "Full predictistic modeling of stock market data: Application to change point problems." European Journal of Operational Research. Elsevier. 2006.

[11] Agarwal, S., K. Deb, "Understanding interactions among genetic algorithms parameters" In: Banzhaf, W., Reaves, C., Foundations of genetic algorithms. Vol 5. 1999.

[12] H. E. Aguirre, K. Tanaka, "Parallel varying mutation genetic algorithms," IEEE transactions, 2002.

[13] D. E. Goldberg, "Sizing populations for serial and parallel genetic algorithms," In: Schaffer, J.D. (Ed), Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kaufmann, Los Altos, CA, pp. 70–79, 1989.

[14] H. Muhlenbein, "How genetic algorithms really work I. Mutation and Hill climbing," Foundation of genetic algorithms II pp. 15-25, 1992. [Online]. Available: http://muehlenbein.org/mut92.pdf

[15] D. E. Goldberg. Genetic algorithm in search, optimization & machine learning. New York: Addison Wisley, 1989.

[16] V. Kapoor, S. Dey, A. P. Khurana, "Empirical Analysis and Random Respectful Recombination of Crossover and Mutation in Genetic Algorithms." International Journal of Computer Application. Special Issue on Evolutionary Computation (1):25–30, 2010. Published by Foundation of Computer Science. Available Online: http://www.ijcaonline.org/specialissues/ecot/number1/1530-133

[17] V. Kapoor, S. Dey, A. P. Khurana "An Empirical Study of the role of Control Parameters of Genetic Algorithms in Function Optimization Problems." International Journal of Computer Application. Volume 31(Number 6): 20-26, October Issue 2011. Available Online: http://www.ijcaonline.org/archives/volume31/number6/3828-5319

[18] R.J. Bauer Jr., Genetic algorithms and investment strategies, John Wiley & Sons, Inc, New York, 1994.

[19] R.J. Bauer, G.E. Liepins, "Genetic algorithms and computerized trading strategies." in D.E. O'Leary, P.R. Watkins (Eds.), Expert Systems in Finance, Elsevier Science publishers, Amsterdam, The Netherlands, 1992.

[20] F. Allen, R. Karjalainen." Using genetic algorithms to find technical trading rules." Journal of Financial Economics 51 (1999) 245-271.

[21] S. Mahfoud, G. Mani, "Financial forecasting using genetic algorithms" Jouranal of Applied Artificial Intelligence. 10(6) (1996) 543-565.

[22] De Jong, K. A., "An analysis of the behavior of a class of genetic adaptive systems". Dissertation Abstracts International 36(10):5140B. (University Microfilms No. 76-9381.) 1975.

[23] Ramon Lawrence, "Using Neural Networks to Forecast Stock Market Prices."

[24] Korczak, J., Roger, P., "Stock timing using genetic algorithms." Applied Stochastic Models in Business and Industry. Pp. 121-134. 2002.

[25] Jin Li, Edward P. K. Tsang, "Improving Technical analysis Prediction: An application of GP." American association of AI. (1999).

[26] Laura Nu´n˜ez-Letamendia, "Fitting control prameters of a genetic algorithm: An application to technical trading system design." European journal of operational research. 2005.

[27] R.J. Kuo, C.H. Chen, Y.C. Hwang "An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network." Fuzzy sets & systems 118 (2001) 21-45