4th WSEAS Int. Conf. on NON-LINEAR ANALYSIS, NON-LINEAR SYSTEMS and CHAOS, Sofia, Bulgaria, October 27-29, 2005 (pp129-134)

Improving Technical Trading Systems By Using A New MATLAB based Genetic Algorithm Procedure

+STEPHANOS PAPADAMOU and ++GEORGE STEPHANIDES
+Department of Economics
University of Thessaly
++Department of Applied Informatics
University of Macedonia
7.5 Km Thessalonikis Asvestohoriou, Dios 3, Postal Code 57010
GREECE

Abstract: - Recent studies in financial markets suggest that technical analysis can be a very useful tool in predicting the trend. Trading systems are widely used for market assessment however parameter optimization of these systems has adopted little concern. In this paper, to explore the potential power of digital trading, we present a new MATLAB tool based on genetic algorithms, which specializes in parameter optimization of technical rules. It uses the power of genetic algorithms to generate fast and efficient solutions in real trading terms. Our tool was tested extensively on historical data of a UBS fund investing in Emerging stock markets through a specific technical system. Results show that our proposed GATradeTool outperforms commonly used, non-adaptive, software tools with respect to the stability of return and time saving over the whole sample period.

Key-Words: - financial markets; prediction; genetic algorithms; non-linear technical rules

1 Introduction

The development of new software technology and the appearance of new software environments (e.g. MATLAB) provide the basis for solving difficult financial problems in real time. MATLAB's vast built-in mathematical and financial functionality, the fact that it is both an interpreted and compiled programming language and its platform independence make it well suited for financial application development.
There have been many studies in the literature concerning the profitability of technical analysis ([1] [2] [3] [4] [5] [6] [7] [8]). However the majority of these studies have ignored the issue of parameter optimization, leaving them open to criticism of data snooping and the possibility of survivorship bias ([9], [10]). Traditionally researchers used ad hoc specification of trading rules. They use a default popular configuration or randomly try out few different parameters and select the best with criteria based on return mainly.
A first trial [11] in implementing a new MATLAB based toolbox for computer aided technical trading presented weak points in the optimization procedure. When the data sets are large and you would like to re-optimize your system often and you need a solution as soon as possible, then try out all the possible solutions and get the best one would be a very tedious task.
In our days, analysts are interested to get a few good solutions as fast as possible rather than the globally best solution. The globally best solution does exist, but it is highly unlikely that it will continue to be the best one. Genetic algorithms (GAs) are better suited since they perform random searches in a structured manner and converge very fast to populations of near optimal solutions. The GA will give you a set (population) of "good" solutions.
The aim of this study is to show how genetic algorithms, a class of algorithms in evolutionary computation, can be employed to improve the performance and the efficiency of computerized trading systems. It is not the purpose here to provide theoretical or empirical justification for the technical analysis. We demonstrate our approach in a particular forecasting task based on the Emerging Stock Markets.

2 Problem Formulation

The last years, there is a growing interest in GA use in financial economics but so far there has been little research concerning automated trading. To our knowledge the first published paper linking genetic algorithms to investments was from Bauer et al [12]. Bauer [13] in his book "Genetic Algorithms and Investment strategies" offered practical guidance concerning how GAs might be used to develop attractive trading strategies based on fundamental information. These techniques can be easily extended to include other types of information such as technical and macroeconomic data as well as past prices.

According to [14], genetic algorithm is an appropriate method to discover technical trading rules. Fernandez-Rodriguez et al [15] by adopting genetic algorithms optimization in a simple trading rule provide evidence for successful use of GAs from the Madrid Stock Exchange. Some other interested studies are [16] presented a new genetic-algorithm-based system and applied it to the task of predicting the future performances of individual stocks; [17] and [18] applied genetic programming to foreign exchange forecasting and reported some success.

One of the complications in GA optimization is that the user must define a set of parameters such as the crossover rate, the population size and the mutation rate. According to De Jong's [19] who studied genetic algorithms in function optimization good GA performance requires high crossover probability (inversely proportional to population size), and a moderate population size. Goldberg [20] suggest that a set of parameters that works well across many problems is crossover parameter = 0.6, population size = 30 and mutation parameter = 0.0333. Bauer [12] performed a series of simulations on financial optimization problems and confirmed the validity of Goldberg's suggestions. In the present study we will perform a limited simulation study by testing various parameter configurations for the trading system tested. We will also provide evidence for the GA proposed by comparing our tool with other software tools.

2.1 Methodology

Our methodology is conducted in several steps. Firstly, we have to implement our trading system based on technical analysis. In developing a trading system, you need to determine when to enter and when to exit the market. If the trader is in the market the binary variable Ft is equal to one otherwise is zero. As position traders we base the majority of our entry and exit decisions on daily charts by constructing a trend following indicator (Dimbeta). This indicator calculates the deviation of current prices from its moving average of θ1 length. The indicators used in our trading system can be formalized as below:

$$Dimbeta_t = \frac{Close_t - MovAv_t(Close, \theta_1)}{MovAv_t(Close, \theta_1)} \tag{1}$$

where Close is the closing price of the fund at time t and function MovAv calculates the simple moving average of the variable Close with time length θ1.

$$MovAv_t(Close, \theta_1) = \frac{1}{\theta_1}\sum_{i=0}^{\theta_1 - 1} Close_{t-i}, \quad t = \theta_1, \theta_1 + 1, \ldots, \mathrm{N} \tag{2}$$

Our trading system consists of two indicators, the Dimbeta indicator and the Moving Average of Dimbeta given by the following equation.

$$MovAv_t(Dimbeta, \theta_2) = \frac{1}{\theta_2}\sum_{i=0}^{\theta_2 - 1} Dimbeta_{t-i}, \quad t = \theta_2, \theta_2 + 1, \ldots, \mathrm{N} \tag{3}$$

If *MovAv(Dimbeta,θ2)* cross upward the Dimbeta then enter long into the market (i.e. buy signal). If *MovAv(Dimbeta,θ2)* cross-downward then close the long position in the market (i.e. sell signal).

Secondly, we have to optimize our trading strategy. It is well known that maximizing objective functions such as profit or wealth can optimize trading systems. The most natural objective function for a risk-insensitive trader is profit. In our software tool we consider multiplicative profits. Multiplicative profits are appropriate when a fixed fraction of accumulated wealth ν>0 is invested in each long trade. In our software no short sales are allowed and the leverage factor is set fixed at ν=1, the wealth at time T is given by the following formula:

$$W_{(T)} = W_o \prod_{t=1}^{T} (1 + F_{t-1} \cdot r_t) \cdot \{1 - \delta |F_t - F_{t-1}|\} \tag{4}$$

where $r_t = (Close_t / Close_{t-1}) - 1$ is the return realized for the period ending at time t, δ are the transaction costs and F_t is the binary dummy variable indicating a long position or not (i.e. 1 or 0). The profit is given by subtracting from the final wealth the initial wealth, $P_T = W_{(T)} - W_0$.

Optimizing a system involves performing multiple tests while varying one or more parameters (θ1, θ2) within the trading rules. The number of tests can quickly grow enormous (Metastock has a maximum of 32,000 tests). In the FinTradeTool [11], there is no limit however the time processing depends on the computer system used. In this paper we investigate the possibility of solving the optimization problem by using genetic algorithms.

Genetic Algorithms (GAs) that were developed by Hollands [21] constitute a class of search, adaptation and optimization techniques based on the principles of natural evolution.

Genetic Algorithms lend themselves well to optimization problems since they are known to exhibit robustness and can offer significant advantages in solution methodology and optimization performance. GAs differ from other optimization and search procedures in some ways. Firstly, they work with a coding of the parameter set, not the parameters themselves. Therefore GAs can easily handle the binary variables. Secondly, GAs search from a population of points, not a single point. Therefore GAs can provide a set of globally optimal solutions. Finally, GAs use only objective function information, not derivatives or other auxiliary knowledge. Therefore GAs can deal with the non-continuous and non-differentiable functions that are actually existed in a practical optimization problem.

2.1.1 Proposed GATradeTool

In GATradeTool, Genetic Algorithm operates on a population of candidate solutions encoded. Each decision variable in the parameter set is encoded as a binary string and these are concatenated to form a chromosome. It begins with a randomly constructed population of initial guesses. These solution candidates are evaluated in terms of our objective function (equation 4). In order to obtain optimality each chromosome exchanges information by using operators (i.e. crossover[1] and mutation[2]) borrowed from natural genetic to produce the better solution.
The objective function (equation 4) is used to provide a measure how individuals have performed in the problem domain. In our case, the most fitted individuals will have the highest numerical value of the associated objective function. The fitness function transforms the raw objective function values into non-negative figures of merit for each individual. The tool supports the offsetting and scaling method [20] and the linear-ranking algorithm [22].
Following genitor selection method [23] we ranked all individuals of a population according to performance based on return. Better performers replaced the poor performers. These candidates were allowed to participate in the crossover and possible mutation. The procedure that recombines promising candidates in order to create the next generation is known as crossover. Finally random mutations [13] are introduced in order to avoid local optima. These steps were repeated until a well-defined criterion is satisfied.

3 Problem Solution

In this section, we apply our methodology in a UBS Mutual Fund investing in emerging stock markets. The data analyzed consists of 2800 observations on daily closing prices of that fund for the period 1/5/98 – 25/6/04. The optimization period is defined between 1/5/98 to 25/6/03. The optimized system was evaluated through the extended period 25/6/03 to 25/6/04.

The optimization problem is set as to determine the optimal lengths of Dimbeta indicator and its moving average for the simple Dimbeta model that will maximize profits. Firstly, the effect of different GA parameter configurations will be studied. More specifically we are interested to measure the effect of the population size and the crossover parameter in the performance of the genetic algorithm based optimization procedure. According to previous research recommendations [20], [12], [24], the population size should be equal to 30 and the crossover rate should be 0,6 (default values). The number of iterations was set to 300 for all simulations. Secondly, we compared the solutions of optimization problem conducted by different software tools in order to measure the validity of the GATradeTool proposed.
Table 1 provides the GA optimization results for different size of populations. The first row of the table shows the best parameters for the Dimbeta indicator and the moving average of Dimbeta. In order to measure the effect of the population size in the best solution we examine a series of different statistics. The solution with the maximum and minimum return, the average return, the standard deviation of these solutions, the time needed for convergence of the algorithm, and an efficiency index calculated by dividing max return solution by the standard deviation of solutions.
By looking in table 1 we can say that as long as you increase the population size the best and the average solutions are higher. However, after a population size of 30 the performance decreased. In order to take into consideration the computational costs involved since increase in population size, we calculate the time needed for solving the problem. Low population size leads to low performance and low completion time. According to the efficiency index the best solution is that given by the population size 20.

Table 1 Population Size Effect

	5	10	20	30	40
Dimbeta/MADimbeta	53/197	71/141	72/135	138/206	203/202
Completion Time	4,68	9,00	17,57	26,66	36,97
Max Return	88,83%	95,35%	121,09%	126,39%	108,87%
Min Return	-40,00%	-0,34%	-53,00%	-87,52%	-15,26%
Avg Return	59,72%	68,25%	69,74%	76,07%	71,77%
St. Dev. Of Returns	29,08%	33,03%	36,97%	42,33%	39,09%
Max Ret./St. Dev	3,05	2,89	3,28	2,99	2,79

Table 2 gives the results of the genetic optimization procedure by altering the crossover rate between 1 and 0.2 for the population size selected from previous table (i.e. 20). The structure of this table is the same like the previous one. For example, when

crossover rate is one, the GA found that the Dimbeta(203,179) had the best performance of 126,39% profit. The second row given the time needed to reach the optimal solution. The next rows give statistics on the evolution process. All configuration studied appear to converge to near optimal solutions, producing large positive profits.
In order to assess the appropriateness of a specific crossover rate, since the models have different initial populations (i.e. initial set of random numbers, the initial conditions) and find different optimal solutions we will examine the stability of the average by using the standard deviation measure. We can see that the most "stable" average population fitness appears for a crossover rate of 60%, this confirms the configuration suggested in the literature.

Table 2 Crossover Effect

	0,2	0,4	0,6	0,8	1
Dimbeta/MADimbeta	105/232	205/234	72/135	216/192	203/179
Completion Time	17,71	18	17,57	18,45	19,1
Max Return	119,14%	105,12%	121,09%	105,12%	126,39%
Min Return	-60,05%	-25,47%	-53,00%	35,00%	-74,84%
Avg Return	74,15%	70,92%	69,74%	67,63%	67,28%
St. Dev. Of Returns	41,65%	39,03%	33,04%	35,39%	35,03%
Max Ret./St. Dev	2,86	2,69	3,66	2,97	3,61

By looking at Table 3 you can compare the results of optimization of our trading system by using three different software tools. The first row gives the result for the GATradeTool against the Metastock and the FinTradeTool. Our proposed software tool (GATradeTool) can solve the optimization problem very fast without any specific restrictions about the number of total tests. The maximum number of test that can be performed in Metastock software is 32000. The FinTradeTool needs much more time in order to find the optimal solution. The solution provided by the GATradeTool, is closed to the optimal solution of the FinTradeTool.

Table 3 Comparison of different software tools

Software Tool	Optimised Parameters (Dimbeta,MovAv(DimBeta))	Total Tests	Completion Time (minutes)	Optimisation Period Return (1/5/98-25/6/03)	Evaluation Period Return (25/6/03-25/6/04)
GATradeTool	(72,135)	-	17,57	121,1%	6,5%
FinTradeTool	(75,129)	39601	67,15	126,4%	11,7%
Metastock	(60,111)	32000	30,3	116,9%	4,5%

The trading systems with the optimum parameters that have been found in period 1/5/98-25/6/03 were tested in the evaluation period 25/6/03-25/6/04. The performance of our trading system has been increased in all software tools. However, the cost of time has to be considered very seriously (column 4).
Figure 2 depicts the evolution of the maximum, minimum and average return across the 300 generations for the Dimbeta trading system (population size 80, crossover rate 0,6). It can be observed that the maximum return has a positive trend. It appears to be relatively stable after 150 generations and moves in the range between 1.2 and 1 (ie. 120%-100% return). For the minimum fitness no pattern seems to exist. For the average population return a clear upward trend can be found in the first 180 generations, this is an indication that the overall fitness of the population improves over time. Concerning the volatility of the solutions, standard deviation of solutions after an increase in the first generations stabilizes in a range between 0.3 and 0.6 providing evidence of a stable and efficient set of solutions.

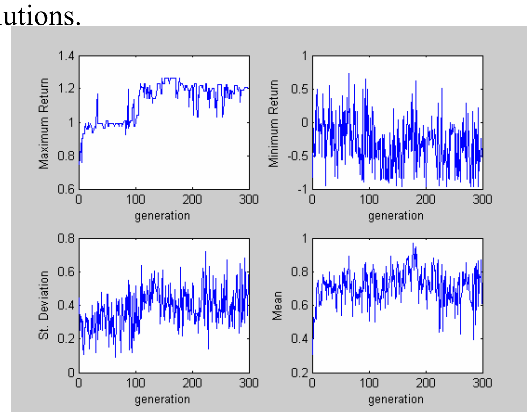

Fig.2 Evolution of statistics over 300 generations

Figure 3 provides a three dimensional plot of the optimum solutions given by the GATradeTool. In axes x and y we have the parameters θ1, θ2 for the dimbeta indicator and its moving average. Axis 2 shows the return of the Dimbeta trading system for the selected optimum parameters. As can be easily understood our tool provides an area of optimum solutions in contrast with the FinTradeTool that provides only the best solution.

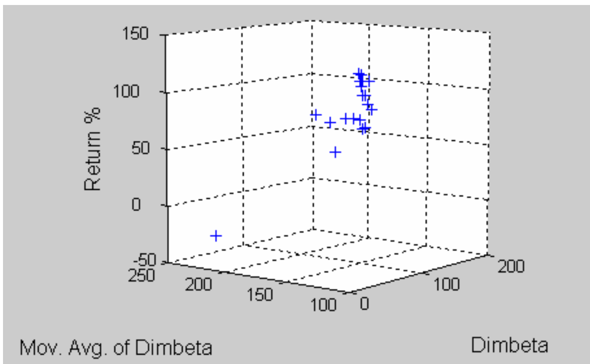

Fig.3 A 3-D Plot of the optimum area

4 Conclusion

Our main objective in this paper is to illustrate that the new technology of MATLAB can be used in order to implement a genetic algorithm tool that can improve optimization of technical trading systems.

Our experiment results show that GATradeTool can improve digital trading by providing quickly a set of near optimum solutions. Concerning the effect of different GA parameter configurations, we found that an increase in population size can improve performance of the system. The parameter of crossover rate does not affect seriously the quality of the solution.

By comparing the solutions of the optimisation problem conducted by different software tools, we found that the GATradeTool can perform better, by providing very fast a set of optimum solutions that present a consistency in all over the evaluation period.

Finally, it would be interesting for further research to test a series of different systems in order to see the correlation between genetic algorithm and system performances. In our days of frequent changes in financial markets the researchers and traders can easily test their specific systems in GATradeTool by changing only the function that produce the trading signals.

References:

[1] Jegadeesh, N. and Titman, S., Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency, *Journal of Finance*, Vol.48, No.1, 1993, pp. 65-91.

[2] Lehmann, B. N., Fad, martingales, and market efficiency, *Quarterly Journal of Economics,* Vol.105, 1990, pp. 1-28.

[3] Werner, F.M., Bondt, D. and Thaler, R., Further Evidence on Investor Overreaction and Stock Market Seasonality, *Journal of Finance*, Vol.42, No.3,1987, pp. 557-581.

[4] Menkhoff, L. and Schlumberger, M., Persistent Profitability of Technical Analysis on Foreign Exchange Markets? *BNL Quarterly Review*, Vol.193, 1995, pp. 189 - 216.

[5] Brock, W., Lakonishok, J. and LeBaron, B., Simple technical trading rules and the stochastic properties of stock returns, *Journal of Finance,* Vol.47, 1992, pp. 1731-1764.

[6] Allen, H. L. and Taylor, M. P., The use of technical analysis in the foreign exchange market, *Journal of International Money and Finance*, Vol.11, 1992, pp. 303 - 314.

[7] Cheung, Y. W. and Wong, C. Y. P., The Performance of Trading Rules on Four Asian Currency Exchange Rates. *Multinational Finance Journal*, Vol.1, 1997, pp. 1 - 22.

[8] Papadamou, S. and Tsopoglou, S., Investigating the profitability of Technical Analysis Systems on foreign exchange markets, *Managerial Finance*, Vol.27, No.8, 2001, pp. 63-78.

[9] Lo, A.W. and MacKinlay, A.C., When are contrarian profits due to stock market overreaction? *Review of Financial Studies*, Vol.3, 1990, pp. 175-206.

[10] Brown, S., W. Goetzmann and S. Ross, Survival, *Journal of Finance*, Vol.50, 1995, pp. 853-873.

[11] Papadamou, S. and Stephanides, G., A New Matlab-Based Toolbox For Computer Aided Dynamic Technical Trading, *Financial Engineering News,* Vol. May/June, No.31, 2003.

[12] Bauer, R. J. and G. E. Liepins, Genetic Algorithms and Computerized trading strategies, *In Expert Systems in Finance*, edited by D.E. O'Leary and P.R. Watkins, Amsterdam, The Netherlands: Elsevier Science Publishers, 1992.

[13] Bauer, R. J. Jr., *Genetic Algorithms and Investment Strategie,* New York, John Wiley & Sons, Inc, 1994.

[14] Allen, F. and Karjalainen, R., Using genetic algorithms to find technical trading rules, *Journal of Financial Economic*, Vol. 51, 1999 pp. 245-271.

[15] Fernández-Rodríguez, F., González-Martel, C. and Sosvilla-Rivero, S., Optimisation of Technical Rules by Genetic Algorithms: Evidence from the Madrid Stock Market, Working Papers 2001-14, FEDEA (ftp://ftp.fedea.es/pub/Papers/2001/dt2001-14.pdf)

[16] Mahfoud, S. and Mani, G., Financial Forecasting Using Genetic Algorithms, *Journal of Applied Artificial Intelligence,*Vol.10,No.6, 1996, pp. 543-565.

[17] Neely, C., Weller, P. and Ditmar, R., Is technical analysis in the foreign exchange market profitable? A genetic programming approach, in Dunis, C. & Rustem, B.(ed.), *Proceedings, Forecasting Financial Markets: Advances for Exchange Rates, Interest Rates and Asset Management,* London, 1997.

[18] Oussaidene, M., Chopard, B., Pictet, O. and Tomassini, M., Practical aspects and experiences – Parallel genetic programming and its application to trading model induction, *Journal of Parallel Computing,* Vol.23, No.8, 1997, pp. 1183-1198.

[19] De Jong, K., An Analysis of the behavior of a class of Genetic Adaptive Systems, Ph.D. diss., University of Michigan, University Microfilms No. 76-9381, 1975.

[20] Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning,* Addison-Wesley, 1989.

[21] Holland, J.H., *Adaptation in natural and artificial system,* University of Michigan Press, 1975.

[22] Baker, J. E., Adaptive Selection Methods for Genetic Algorithms, *In Proceedings of the first International Conference on Genetic Algorithms*, 1985, pp. 101-111.

[23] Whitley, D., The Genitor algorithm and selection pressure: why rank-based allocations of reproductive trials are best *In Proceedings of the third International Conference on Genetic Algorithms*, 1989, pp. 116-121.

[24] Markellos, R.N., Backtesting trading systems, *Journal of Computational Intelligence in Finance,* Vol.5, No. 6, 1997, pp. 5-10.

Footnotes

1 Arithmetic single-point crossover, involves randomly cutting two strings at the same randomly determined string position and then swapping the tail portions. Crossover extends the search for new solutions in far-reaching directions.

2 Mutation is a genetic operation that occurs with low frequency and alerts one character in a particular string position. For example a 0 in a string could be altered to 1, or vice versa, through mutation.

Acknowledgements:

This research paper was part of the postdoctoral research of Dr. S. Papadamou that has been funded by IKY Greek State Scholarships Foundation.