An Implementation of Genetic Algorithms as a Basis for a Trading System on the Foreign Exchange Market

Andrei Hryshko
School of Information Technology & Electrical Engineering,
The University of Queensland, QLD, 4072,
AUSTRALIA
dushenka@itee.uq.edu.au

Tom Downs
School of Information Technology & Electrical Engineering,
The University of Queensland, QLD, 4072,
AUSTRALIA
td@itee.uq.edu.au

Abstract - Foreign Exchange trading has emerged in recent times as a significant activity in many countries. As with most forms of trading, the activity is influenced by many random parameters so that the creation of a system that effectively emulates the trading process will be very helpful.

In this paper we try to create such a system with a Genetic Algorithm engine to emulate trader behaviour on the Foreign Exchange market and to find the most profitable trading strategy.

1. Introduction

The Foreign Exchange Market (FOREX) is an interbank market that was created in 1971 when international trade started to use floating rather than fixed exchange rates. FOREX was an integration of deregulated domestic stock markets in leading countries. Since the time of its creation, rates of currencies have been determined on the market based on demand and supply in regards to each currency. [1]

There are two basic approaches to market analysis: fundamental analysis and technical analysis.

Fundamental analysis focuses on the economic forces of supply and demand and determines whether prices move higher, lower or stay almost the same. It examines all relevant factors affecting the price in order to determine the intrinsic value of the market. Intrinsic value means worth based on the law of supply and demand.

Technical analysis is the practice of trying to forecast market prices by examining trading patterns and comparing the shape of current charts to those from the past.

Both of these approaches to market forecasting try to solve the same problem i.e. to determine in what direction prices will move. The difference between fundamental and technical analyses is that the first one explores the causes of movements on the market and the second one explores the effect of these movements.

Fundamental analysis is generally concerned with longer-term trends than technical analysis so, given that most traders on FOREX are intraday traders they use just technical analysis and do not take into consideration the fundamental method. Traders who base decisions upon technical analysis are known as *technicians* and it is for their activities that we are seeking to develop an advisory tool based upon Machine Learning.

Technicians are divided into two groups: traditional *chartists* and *statistical* technicians. For a chartist, the analysis of charts is the primary working tool. Charting remains largely subjective and the success of this approach depends on the skill of the individual chartist.

In contrast, the statistical technician approaches the market in terms of developing a mechanical trading system. Such systems are then programmed into a computer that generates the mechanical trading advice to "buy" or "sell".

2. Statistical approach to technical analysis

The main instrument employed by statistical technicians is the set of available indicators which helps traders to discover trends, trend reversals and other fluctuations.

Different indicators are used for different purposes. Some indicators work better when the market has a strong trend, some when the market is neutral. The main indicators are fully described in [2]. To illustrate the use of indicators consider the following two examples.

(i) The simplest of all indicators is the Moving Average (MA) that shows the average value of prices during a defined period of time. For the n-day MA the following formula applies:

$$MA = \frac{P_1 + \ldots + P_n}{n} \qquad (1),$$

where P_i is the price i-1 days previously.

A buy signal is produced when a shorter average crosses above the longer one. When the shorter average moves below, a sell signal is generated. This technique is called the *double crossover method.*

As an illustration Figure 1 shows the bar chart of Deutsche Mark exchange rate and two moving averages:

the 40-day MA (dashed line) and the 9-day MA (solid line).

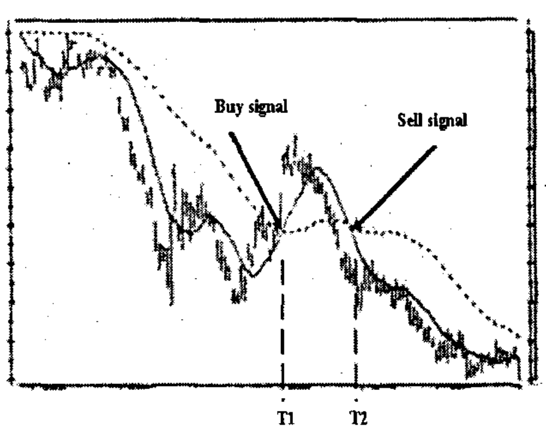

Figure 1: The moving Average Indicator; 40 day MA (dashed line); 9 day MA (solid line)

At time T1 the 9-day MA crosses above the 40-day MA and produces a 'buy' signal. At time T1 the 9-day MA crosses below the 40-day MA and produces a 'sell' signal.

A slightly more sophisticated indicator is the Moving Average Convergence/Divergence (MACD), an example of which is shown in figure 2. It consists of two lines: the solid one is called the MACD line and the dotted one is called the signal line. The MACD line is the difference between two exponentially smoothed moving averages of closing prices and responds very quickly to trend movements. The signal line is the exponentially smoothed average of the MACD line and it responds more slowly to trend movements.

To calculate the MACD we need to compute the n-day Exponential Moving Average (EMA):

$$EMA = kP_{today} + (1-k)EMA_{yesterday} \qquad (2),$$

where P_{today} is today's price and k is usually set to $2/(n+1)$;

Thus, for example, if we want to calculate the 10-day EMA we need to

1. Calculate the 10-day MA for the first 10 days of historical data. This gives the starting for using equation (2) where the 10-day MA is set equal to the 10-day EMA.

2. On the 11-th day calculate EMA_{11} using the equation (2).

3. Repeat step 2 in the obvious way until we get the EMA for the present time.

Then, the expression for calculating the MACD is

$$MACD\ Line = EMA_n - EMA_m$$

and it represents the difference between the n-day and the m-day exponentially smoothed moving averages.

The Signal Line is given by the p-day exponentially smoothed average of the MACD line.

Buy and sell signals arise when these two lines cross each other. A crossing by the MACD line that takes it above the Signal line is a buy signal and a crossing by the MACD line that takes it below the Signal line is a sell signal.

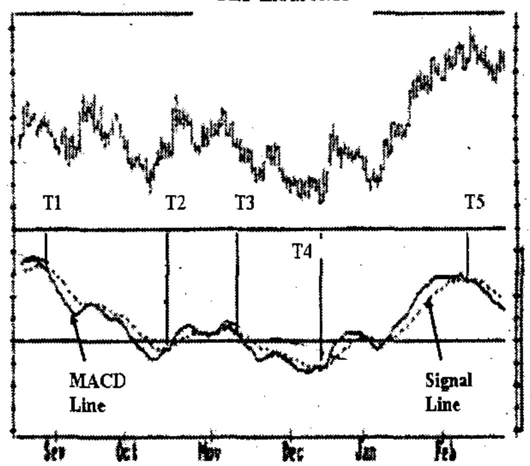

Figure 2. The Moving Average Convergence Divergence System.

Figure 2 shows five MACD trading signals on a chart of the S&P index. At times T1, T3 and T5 there are 'sell' signals because the MACD line crosses below the signal line. At times T2 and T4 the MACD line crosses above the signal line so the system gives 'buy' signals.

3. Machine Learning on Financial Markets

Usually traders start with some concepts based upon indicators and then turn those concepts into a set of objective rules. The rule creating process requires the trader to choose which indicators to rely on and to use his own subjective opinion and intuition to define the rules on how to interpret the indicator signals. Then he has to program the rules to provide technical analysis software. This is a difficult task for experienced traders, not to mention beginners. The market is changing all the time so that different rules and concepts work in different situations, and it often happens that a system that works well at a given time is incapable of giving good advice two hours later. Hence a trader has to determine the times at which his program should be changed and obviously he cannot modify his program on-line. Another problem for the trader is that different indicators are successful on different financial markets. For example, indicators that are profitable on one set of exchange rates could lead to catastrophic losses on another. Therefore a trader must tailor his strategy to a particular situation.

The approach described in this paper is to design a system engine based on machine learning and embed it into a trading system. This system will draw upon available information to determine the optimum strategy for the trader. Moreover, unlike the human trader, it will work on-line and so will update its parameters over time to achieve the highest returns. This system will take decisions and predict the future market based on a combination of different market models. The system will automatically recognise the condition of the market based on the simultaneous examination of signals from all indicators (instead of examination of indicator signals one by one).

We present results from a very basic implementation of our approach and show that it is only slightly inferior in performance to the method recently published in [4]. We then describe a more elaborate extension to our approach that is expected to provide substantial improvement in performance. It is intended that results from this extension will be available at the conference.

4. Performance Criteria.

Ultimately the task is to maximise some relevant measure of trading system performance. There are many different ways to estimate the performance of a trading strategy. This can be done in terms of profit, wealth or performance ratio. A widely used measure in classical finance is the Sharpe Ratio. [3] It is defined as excess return (earned reward) divided by risk (measured in terms of standard deviation of return). From our point of view this measure has a significant disadvantage because people trading on the financial market are usually concerned about how much they can lose rather than probability of loss. So we propose to use the Stirling ratio which is excess return divided by maximum drawdown, where maximum drawdown is the maximum losing streak that has previously occurred. The equation for the Stirling Ratio is

$$S = \frac{Return}{Maximum\ drawdown}$$

The fraction of positions that have been profitable is also used as a performance ratio. This is defined as:

$$F = \frac{No.\,of\,winning\ positions}{No.\,of\ positions}$$

and will also be used in computing our results.

5. Model Architecture

The general principle of trading is very simple – buy when the price is low and sell when the price is high. If a trader follows this strategy he also gains a profit. The problem is how to determine the times to sell and buy. A trader's aim is to recognise the time when to enter into the market and when to exit the market. This means that there should be a trading model which consists of two parts, an entry strategy and an exit strategy. Figure 3 illustrates the principles of trading and the use of the two strategies.

Before discussing Figure 3, it will be helpful to explain a few terms employed in the trading arena. To take a *short position* means to sell a security, contract or commodity and to take a *long position* means to buy it. A neutral position is a situation where the trader takes neither a long nor a short position.

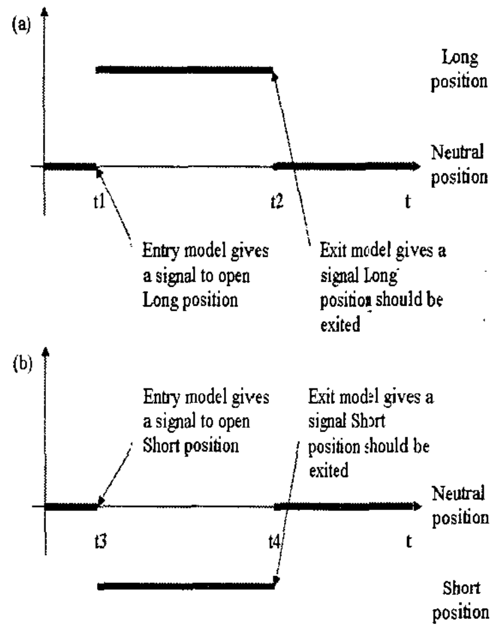

Figure 3: Trading Strategies

Plot (a) in Figure 3 illustrates the situation when a trader at time t1, based on his entry strategy takes a long position and holds it until time t2. Then at time t2, based on his exit strategy, he terminates the long position (sells the currency he bought at time t1) and takes a neutral position.

Plot (b) illustrates the opposite situation where the trader at time t3, based on his entry strategy, takes a short position and holds it until time t4. Then, based on his exit strategy, he terminates the short position (buys the currency he sold at time t3) and takes a neutral position.

In our context, we will use a genetic algorithm (GA) that seeks a function that converts the market state to an action that is either to enter the market or to exit the market. If the action is to enter the market the function must indicate whether a short or long position should be adopted. In the GA, trading rules are represented as binary trees where the terminals are the individual indicators defining the market state and the non-terminals are Boolean functions. [4]

For instance, consider a trader who is buying and selling a particular foreign currency. Under his entry strategy the trader adopts a position depending on the value of the root node (ACTION), where TRUE represents a long position in the foreign currency and FALSE represents a short position.

The system rule has the following form

INDICATOR | CONDITION (true or false) | CONNECTOR (Boolean operator) | INDICATOR | CONDITION | INDICATOR | ... | ACTION (buy or sell).

For example a trader could have the following rule:

{Stochastic | TRUE | Momentum | FALSE | RSI | TRUE | CCI | TRUE | Williams | TRUE | BUY}.

For his exit strategy the rule has the following form

INDICATOR | CONDITION (true or false) | CONNECTOR (Boolean operator) | INDICATOR | CONDITION | INDICATOR | ... | EXIT

The exit rule differs from the entry rule in that the output of the exit rule is determined *a priori* whereas in the entry rule the output can be either sell or buy and so is essentially random.

Table 1 and Table 2 show examples of binary strings for entry and exit rules respectively. The strings will obviously be significantly longer when a large number of indicators have to be coded.

SELL	MACD	CONNECT	Momentum	CONNECT	Stochastic
0	0	01	0	00	1
	FALSE	OR	FALSE	AND	TRUE

Table 1

EXIT	SRC	CONNECT	RSI	CONNECT	CCI
	1	10	0	00	0
	TRUE	XOR	FALSE	AND	FALSE

Table 2

Table 1 illustrates the situation (rule) when the trader should adopt a short position. According to the rule If (the Stochastic indicator signal is 'buy' AND the Momentum indicator signal is 'sell') OR the MACD indicator signal is 'sell' Then the short position should be adopted.

Table 2 illustrates the exit rule. According to the rule If (the CCI indicator signal is 'sell' AND the RSI indicator signal is 'sell') XOR the SRC indicator signal is 'buy' Then the short position adopted previously according to rule in the Table 1 should be exited.

In a typical implementation, a population of 50 rules of each strategy is generated randomly. Then we randomly combine 50 pairs consisting of one entry model rule and one exit model rule. Evolution then proceeds by generating a new population from this. Pairs are ranked and are chosen in proportion to their rank to be involved in crossover and mutation. Crossover and mutation occur only between rules of the same type (entry or exit). Crossover combines two rules to form two "children" by swapping sub-trees. Mutation makes a random change (with probability 0.02) in a single rule to create a new one. This process continues for 100 generations after which the single best pair is chosen to be the output of the genetic algorithm. The best pair means the pair that has given the maximum performance ratio.

With regards to fitness evaluation the in-sample period (training period of the whole system) is broken into two parts, a training sub-period and an evaluation sub-period. The evaluation sub-period is used to evaluate appropriate individuals within the GA's population of potential solutions. The Stirling Ratio and fraction of winning positions over the evaluation period are used as the fitness function of the GA.

The following outlines the procedure of applying Genetic Algorithm in more detail:

- Initialise population of rules of entry and exit models.
    - Generate number of indicators N (less or equal to total number of available indicators) that will be included into consideration.
    - Generate N binary numbers (indicators conditions, TRUE or FALSE)
    - Generate connectors (OR, AND or XOR)
- Fitness evaluation
    - Testing trading performance of every pair of rules over historical data
    - Use Stirling ratio to choose which pairs will be participating on crossover and mutations.
- Crossover
    - Best s% pairs of rules are considered for crossover
- Mutation
    - Mutation rate of 2%

The pair of rules generated by this procedure is then used for taking decisions. If the chosen strategy makes a loss greater than 1% of trade capital or a loss for three consecutive weeks we restart the procedure with a new initial population of rules.

6. Previous work

Dempster and Jones in [4] also applied GAs to trading on the Foreign Exchange market, but their system differs from ours in that their approach is based upon two-way trading with equivalent credit in both traded currencies. They assumed a fixed-credit line in US dollars with a notional principal of one million to buy British pounds and the equivalent amount in pounds to buy US dollars. Our approach aims to discover the most profitable one-way trading rules because this is a more realistic situation.

Another major distinguishing feature of approach in [4] is that it uses only an entry model for the system and therefore does not have a neutral position for the trader unless the trader has lost more than 100 points and the position is closed by default. "A trade is entered (long or short) when a (BUY or SELL) rule gives a signal. Trades are exited either when the cash management rule is activated or the trade is reversed when a SELL (BUY) rule is activated whilst a long (short) position relative to the purchased currency is held" [4]. In contrast, the trading model presented in this paper moves into neutral mode when the exit strategy is applied.

A comparison of the performance of our method with that of the method of [4] is given in the next section.

7. Numerical results

Five months of data were used from 1 March, 1996 to 2 August, 1996 for the USD. As is common in financial analysis, the data were split into two parts, an "in-sample" period and an "out-sample" period. The in-sample period is used for optimising the performance of the trading system and determining the best strategy. The out-sample period is used for testing the strategy that has been arrived at through GA.

Results are presented in Table 3. Results obtained by the approach in [4] are shown in Figure 4 for trading the British Pound against the US Dollar.

Currency GBP/USD. Crossover rate, s=0.5. Number of indicators, N=10		
Performance	In-sample 1/03/1996-1/05/1996	Out-sample 2/05/1996-2/08/1996
Fraction	0.56	0.47
Stirling Ratio	-0.02	-0.093

Currency USD/DEM. Crossover rate s=0.5. Number of indicators, N=10		
Performance	In-sample 1/03/1996-1/05/1996	Out-sample 2/05/1996-2/08/1996
Fraction	0.49	0.42
Stirling Ratio	-0.033	-0.15

Table 3

Table 3 illustrates statistical results for the GBP/USD and USD/DEM currency pairs. These results were obtained using the following steps:

1) Take a long or a short position based on an entry rule until the corresponding exit rule gives a signal to exit the position.
2) When the position is closed, the return and maximum drawdown are calculated based upon exchange rates at the moment of time the position was taken and at the present time. If the return is positive, the position is considered to be profitable and the fraction should be increased.

The first column illustrates the results for the in-sample period where the GA-optimised system achieved slightly negative results. Table 3 illustrates results only for the most profitable couple of exit and entry rules and for the last iteration of the GA. The Stirling Ratio is slightly less than zero which indicates a small loss for both pairs. The fraction of winning positions for the USD/DEM pair is less than a half and indicates that 49% of positions resulted in profit. For the GBP/USD pair we have a fraction of more than a half which illustrates that the majority of positions resulted in profit.

The second column in the tables illustrates the results of the most profitable couple of rules for the out-sample period. This shows that the fraction of winning positions and the Stirling ratio for the out-sample period are slightly worse than for the in-sample period, as we would expect. (The in-sample period provides the data on which the system is trained and the out-sample period provides test data).

Figure 4 shows the convergence of the top 50 strategies obtained by the approach in [4]. It can be seen that the profitability of the top strategies increases with each iteration and that finally the Stirling ratio of the most profitable rules is close zero.

This indicates that the method in [4] outperforms our preliminary method because our result is -0.03. The major difference in our approach is that we use two rules (entry and exit) whereas the approach in [4] uses a single rule that from our point of view is less realistic.

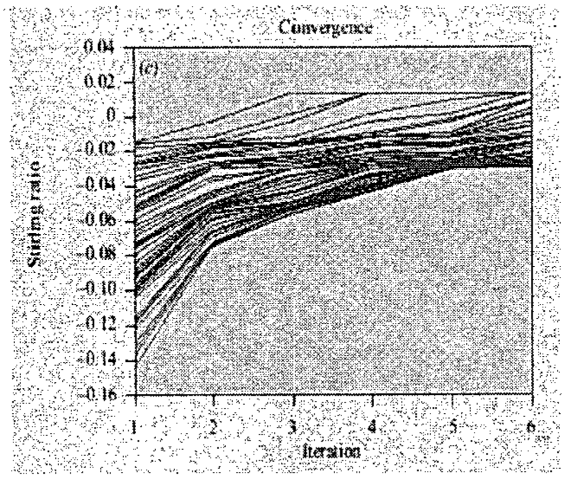

Figure 4. Trading results using approach in [4]

We now outline an algorithm that we expect to give more accurate predictions and better system performance. Results are not yet available but will be presented at the conference.

The system engine employs both GA and the Reinforcement Learning (RL) technique.

8. Reinforcement Learning

In a reinforcement learning task, an agent faces the following problem: he has to explore his environment to make decisions in different situations but he does not have a complete knowledge about this environment. The only feedback that the agent has from environment is the so called reinforcement signal. The agent's behaviour should choose actions which increase the long-term sum of the reinforcement signals. [5]

Besides the reinforcement signal an agent also receives the current state of the environment (in the form of a vector of observations).

Figure 5 illustrates the principles of Reinforcement Learning. At the moment of time t the agent receives inputs x_t (vector of observations) and r_t (reinforcement signal) and based on this he chooses an action, a_t. The state of the environment s_t receives the action a_t from the agent and goes to state s_{t+1}. At moment t+1 the agent receives inputs x_{t+1} and reinforcement r_{t+1}.

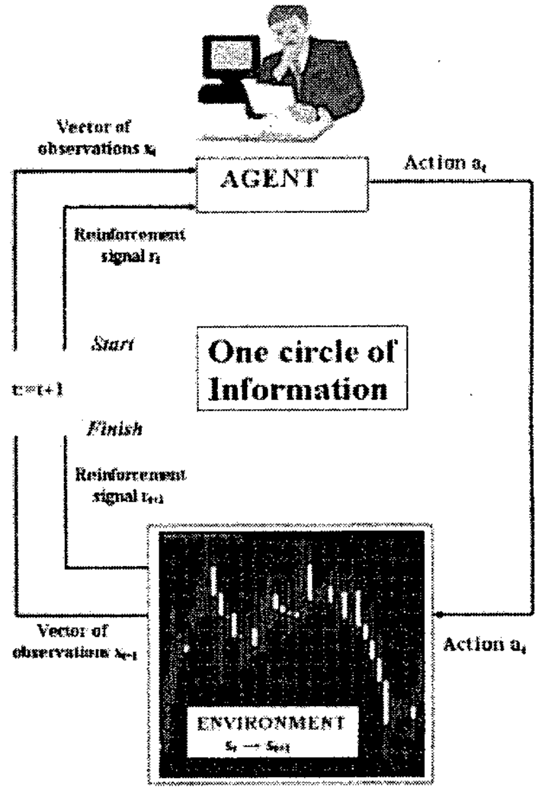

Figure 5. Reinforcement Learning

On the Foreign Exchange market a trader does not have complete knowledge about the environment and he does not know *a posteriori* what position should be adopted to obtain a profit. The trader has only information about losses and wins and this can be considered as the reinforcement signal. For this reason RL is a highly suitable engine for a trading system on the Foreign Exchange market. We should point out that RL has already been used for training a trading system in another context. [6]

9. Reinforcement Learning system algorithm

The stages of the system being developed are:

1. Search for the best "working set" of indicators (number of indicators N=2). Get the optimal performance with the two indicators. We denote this P_2. The idea of the first step is to choose two indicators, from the full set of available indicators, that are the most profitable.

2. We now have the set of N most profitable indicators. Add the indicator that provides best performance with N:=N+1 using the GA. Determine the optimal performance P_{N+1}. Compare P_{N+1} with P_N.

3. If P_{N+1} is less than or equal to P_N this means we cannot improve system performance by adding indicator N+1. In this case go to step 4. If P_{N+1} is greater than P_N this means system performance *can* be improved by adding at least one indicator. In this case go to step 2.

4. The subset of N indicators is fed to the Reinforcement Learning system. Signals from indicators are considered as the state of the environment.

5. The RL system makes an on-line adjustment to system parameters so as to maximise performance. This adjustment effectively modifies the systems response to different combinations of indicators signals. Thus the aim of RL is to interpret the chosen group of indicators in the most profitable way.

6. Go to Step 1 and repeat all steps to find the best working set of N indicators to be fed into the RL system. Note that because this is an on-lin learning system .this process is continued for the life time of the system.

10. Conclusion

In this paper, a GA has been used as an optimiser of trading strategies on the Foreign Exchange market. Here we have shown that a GA is capable of searching over a given set of trading rules and optimising trading performance. It was shown that the optimised indicator rules are able to achieve moderate in-sample gains but failed to make positive (or null) profit during the out-sample period.

Nevertheless, the paper has shown how Machine Learning techniques can be employed for developing strategies for Financial Markets. The next step is to consider Reinforcement Learning techniques together with a Genetic Algorithm and we are confident that this will provide superior performance. We hope to demonstrate this at the conference.

11. Literature

[1] E. Carew and W. Slatyer "Forex : the techniques of foreign exchange", Publisher Sydney : Allen & Unwin, 1989.

[2] J. Murphy "Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications", Prentice Hall Press, 1999

[3] E. Derman "The perception of time, risk and return during periods of speculation", Quantitative-Finance. Aug. 2002: 2(4): 282-96

[4] M. Dempster and C. Jones "A real-time adaptive trading system using genetic programming", Quantitative Finance 1 (2001) 397-413

[5] D. Bertsekas and J. Tsitsiklis "Neuro-Dynamic Programming", Belmont, Mass. Athena Scientific, 1996.

[6] J. Moody and M. Saffell "Learning to Trade via Direct Reinforcement", IEEE Transactions on Neural Networks, Vol 12, No 4, July 2001