An Evolutionary Approach to Modelling the Behaviours of Financial Traders

Sonia Schulenburg
Artificial Intelligence,
Division of Informatics,
The University of Edinburgh
Edinburgh EH1 1HN, UK
sonias@dai.ed.ac.uk

Peter Ross
Artificial Intelligence Applications Institute,
Division of Informatics,
The University of Edinburgh
Edinburgh EH1 1HN, UK
peter@dai.ed.ac.uk

Abstract

We begin this paper by describing our rationale and overall design of an artificial stock market model based on the evolution of a number of economic agents, each of which represents a group of prototypical traders making investment decisions under a well defined market environment. Then, we continue by summarising the simulation details and describing the type of results obtained from implementing the proposed multi-agent system, which consists of three heterogeneous adaptive agents. The idea behind the model is that one agent of each type stands for the aggregate effect of all such traders, and it is a hypothesis of this work that these aggregates are simpler than individual agents would need to be. In addition, it is also possible to choose to vary what a specific type sees according to its nature. Each type of agent, and its effect, is evolved. The agents' decision-making processes are represented by learning classifier systems, effects are self-adapted.

First we present the outcome of a market scenario in an adaptive environment (genetic algorithm enabled) with the agents trading with real data over a period of time. We show that the agents are not only able to survive, but also to perform better than (a) the hypothetical case of leaving all the money in the bank at a fixed compounded rate of interest during the same period of time, and (b) the buy-and-hold strategy, regardless of the fact that there is a commission incurred for every transaction performed. More detailed observations of these results, such as evolution parameters are explained in this paper. Then, we test the agent's performance with these sets of evolved strategies using not previously seen data, with no genetic action involved. Results are consistent with the previous scenario. We continue by analysing further learning over this test set and also comparing results with various genetic algorithm scenarios over the training set. As this is an on-going research, results to come might lead to new insights about how adaptive computation is actually accomplished in natural systems, and further understanding of how we might model the collective behaviour of a large group of agents by a small group of prototypical agents.

1 INTRODUCTION

The change over time of stock prices, foreign exchanges, interest rates, and other factors in financial markets present a very interesting prediction problem that has been widely addressed when considered from a time series perspective. It has, although, been impossible to solve this task perfectly. Most of these predictors can not be trusted under certain circumstances because the underlying rules of the system continuously change. Even those models which are updated very rapidly with new market moving information are always learning from past examples. Such non-stationarity affects all attempts to forecast markets. A second approach, the one which is presented here, would be to be able to model this kind of behaviour, and even the actual behaviour, of a financial market from the agent's perspective.

Time series analysis, e.g. using recurrent or complex-valued neural networks, are problematic because they only model the data, not the causes, but the causes change with time. There is a dynamic system that governs the global evolution of the market and our understanding of the mechanisms that generate this time series is still very limited.

One of the most common problems we face today is an inability to recognise fundamental changes in direction in the causes of the time series analysed. So they are unreliable when there are sudden financial changes in trend [3]. Trying to approach this problem differently by 'viewing' not the time series, but rather the causes of such moves, motivates the development of a system capable of modelling the behaviour of traders themselves, so that market behaviour is a simple consequence of their actions.

The idea of our model is to have only one agent of each type. This agent stands for the aggregate effect of all such traders. For each trading cycle, the decision-making process is represented by Michigan-style classifier systems running simultaneously for each trader. It is a hypothesis of this work that such aggregates are simpler than individual agents would need to be. They have a variable-sized effect on the market; this expresses the fact that if a real trader is successful, others will tend to emulate his/her approach and so the impact of that type on the market increases.

In this paper, each type of agent is represented by a learning classifier system (LCS). However, different types receive different environmental messages. The agents do not communicate or interact with each other. Here, we report experiments that use 10 years of historical data about the stock price of Merck & Co.; as it commonly happens to stocks that have been in the U.S. market over the past decade, this performed very well over the period. We have also studied our system's performance on a contrastive performer, Forest Oil, whose price fell significantly over the period (at least until recently when oil prices began to rise again). The trading agents also managed to perform well in those adverse market conditions, but space does not permit us to report those results here.

2 THE MODEL

Thinking in terms of an agent's capabilities, we have embedded two processes in our agents. The first one consists of a very straight forward deductive approach: making simple deductions from raw market data by calculating various price moving averages, first differences in prices, volume ratios, etc. to generate each agent's pool of beliefs –a different one for every agent. This corresponds to the idea that traders react differently given the same information, they tend to take into consideration different facts about a current market situation. As for the second one, once the agent has produced its own derived information set, the classifier system comes into place to follow each trader's inductive way of reasoning.

2.1 MARKET STRUCTURE

We set up an asset market model similar to that of [1], being as simple as possible in terms of its economic components and using ingredients (and notations) from existing models wherever possible. Time is discrete, indexed by t and represents one trading cycle equal to one day in real terms. In addition, there are other components of the market such as:

- Three heterogeneous **adaptive agents**. Rather than choosing or adjusting numbers of agents in the model, the idea is that the single instance of a type will have a variable-sized effect on the market. This corresponds to the property that if real traders see a strategy that wins, they will tend to try to adopt it, and vice versa.
- One **non-intelligent agent** which represents the so called "buy-and-hold" strategy, which is often regarded as the best strategy to employ for the long run. This agent is of great relevance because we need to compare our trader's performances against a strategy which works very well for highly performing stocks such as the one used here.
- The **Information Set**. This is the available raw data about the market, as well as data which has been processed in various ways. It includes basic daily information about the stock such as its current price, volume of transactions, splits and dividends, and some derived information such as price differences, moving averages, current standing of the buy-and-hold strategy and the bank investment. The agent types are distinguished from each other by the fact that they receive different sets of information daily (environmental messages, in classifier system parlance).
- Two assets traded: a **risk free bond** with infinite supply paying a fixed interest rate, and a **risky stock** issued in N units where payments in the form of dividend and splits apply.

2.2 DECISION MAKING

The goal of the artificial agents is to maximise their profits by making investment decisions regarding their current portfolio. Every day, they have a choice of (i) leaving their money in the bank, in which case there is a fixed interest rate paid on a daily basis equivalent to 8% annually, or (ii) they can buy or sell a stock, which in the following simulations will be represented by Merck & Co.

The decision represents a specific action to take on the current period: buy, sell or hold. But this is only part of the story. This single action, in turn, is linked to a second decision which represents the proportion to trade. If the agent decided to buy, it must also decide "how much" of its current cash available is going to spend in shares; or vice versa, if the decision is to sell, it must specify "how many" shares from its current holdings is going to sell on that day.

2.2.1 Representation

The framework for representing the adaptive agents relies in the following three components, better described in [5],[6]:

- The performance system, consisting of (i) detectors for extracting information from the environment, (ii) classifiers to represent the agent's capabilities for processing the environment and (iii) effectors, through which the agent acts on its environment.
- Credit Assignment Algorithm. For situations in which many rules fire at the same time, it is responsible of strengthening rules that set the stage for later more rewarding activities.
- Rule Discovery Algorithm through which plausible hypotheses are generated and past experience is incorporated.

In order to keep the model as simple as possible, we have introduced only three types of traders in the market and they all receive different sets of market information. We abbreviate the first type as **Tt1**, the second as **Tt2**, and finally the third one as **Tt3**.

Each type of agent controls its position by using and evolving a set of classifiers. These classifiers are used to match binary conditions in the market, which requires predefining a set of binary states or environmental messages for each agent that can be used when making decisions. All three messages include the first difference in price, meaning that all agents see whether the current price of the stock is higher than the previous day's, followed by three basic sets of market information, one for each type of agent.

The first set contains **price statistics and moving averages**, based on standard moving average types of trading rules, the second set contains **volume statistics**, and the third set incorporates apart from the price and volume, some other **news** such as the accumulated wealth of both: the buy-and-hold strategy and the bank's, and a recollection of its past action.

The market states are summarised in a binary state vector of fixed length for each trader type. Tt1's state vector is seven bits long, while for Tt3, the length is only five. This is due to the fact that each trader type receives different information about the market. The traders are thus limited to put their position based on the state variables we give them, but they are capable of ignoring any of these state variables. Each element of the market state vector corresponds to whether some condition is true or false.

Table 1 shows the meaning of each position for one type of trader during the current trading period t. The actual value of each cell is 1 if the condition is satisfied. For example, bit number 1 is 1 if the price today is higher than yesterday's: P_t is the price on day t, P_{MA5} is the five-day price moving average, etc., and the last two cells indicate whether the current price is the highest so far $P_{highest}$, or the lowest ever P_{lowest}.

Table 1: Environmental message for Trader Type 1

Bit Number	Meaning
1	$P_t > P_{t-1}$
2	$P_t > 1.2 * P_{MA5}$
3	$P_t > 1.1 * P_{MA10}$
4	$P_t > 1.05 * P_{MA20}$
5	$P_t > 1.025 * P_{MA30}$
6	$P_t > P_{highest}$
7	$P_t < P_{lowest}$

Similarly, the second and third trader's market state vectors are summarised on table 2 and table 3 respectively.

Table 2: Environmental message for Trader Type 2

Bit Number	Meaning
1	$P_t > P_{t-1}$
2	$V_t > V_{t-1}$
3	$V_t > VMA20$
4	$V_t > V_{highest}$
5	$V_t < V_{lowest}$

The action is a bit string of length 4 indicating the decision of whether to buy, sell or hold possessions on the current day. The first bit represents a buy (1) or sell (0) signal, followed by three more bits representing

Table 3: Environmental message for Trader Type 3

Bit Number	Meaning
1	$P_t > P_{t-1}$
2	$V_t > V_{t-1}$
3	$Shares_t >$ Buy and Hold $Shares_t$
4	$W_t > W_t$ Bank
5	$W_t > W_t$ Buy and Hold
6	Tt3 action at (t-1)

which exact proportion of the agent's shares or available cash are to be traded. For example, an action 0 001 means sell 15% of shares owned, while 1 010 means buy 30% of cash available. The options are 15%, 30%, 45%, 60%, 75%, 90% and 100%, although it is possible to scale them into different figures.

Although it is well known that economic and financial models are mainly based on continuous real valued data, we believe that in this model it is not necessary to have to use continuous entities explicitly in the trader's rules. This binary-encoded information seems to be enough to capture the causal reasoning in such simplified models, and has the added benefit of being easy to process. Associated with each classifier rule are the usual parameters such as the classifier's strength, bid, ebid, match flag and specificity. More detailed definitions of these parameters can be found in [4].

2.2.2 Learning and Rule Evolution

There are two levels of learning in the model:

The **first level** happens rapidly as the agent learns which of his strategies are accurate and worth acting upon, and which ones should be ignored. This happens during the auction among currently matched classifiers in the apportionment of credit algorithm.

On each trading cycle zero or more rules can match the current state of the market, but only the best will be used in that period as it relates to a specific signal. The credit-distribution system used in the present model is a simplified bucket brigade algorithm in which a payment equal to the bid of the current winner is transferred to the old winner only, no chains are implemented because we are not interested in rewarding sequences of classifiers. For each agent there exists an option of setting it on or off.

The **second level**, occurs after several trading periods have gone by, when the structure of these mental strategies will be modified by the use of a genetic algorithm which will replace some of the worst performing rules by new ones created with useful pieces of good rules.

In general terms, a genetic algorithm (GA) is activated to allow the agent to perform operations such as (a) specialisation, by transforming general rules into more specific rules, (b) diversification, by introducing more heterogeneity in the rule set, (c) creation, by adding new rules into the system when an incoming message does not match any rule and (d) generalisation, by creating offspring from parents with high fitness through crossover.

A proportion of the population is selected for reproduction during a given genetic algorithm invocation. It is usually set to 20% of the entire population of classifiers. Using roulette wheel selection, a pair of mates is selected, where crossover and mutation take place in the normal way and the replacement candidates are chosen from a low-performance subpopulation. Since the search is for a well adapted set of rules, we use crowding replacement to choose the classifiers that die, inserting new offspring in their place on the basis of similarity. For more details on how the GA works, refer to [4].

2.3 THE TRADING PROCESS

Each **agent** represents: (i) a proportion of the total wealth, and (ii) the collective decision making behaviour of a large number of traders of its type, given the information pertaining to that subset. It buys, sells, or holds its possessions and adapts by receiving feedback from the changing environment by monitoring, updating and evolving new procedures. The idea is that agents continually form individual, hypothetical expectational models of theories of the market, test them, and trade on the ones that predict best.

An agent might interpret a certain state of the market (events) according to its past performance. Experience plays an important role as it is transferred to the future. Thus there is a dynamical approach in the sense that future behaviour is affected by past performance importing feedback (new observations) from the environment. This inductive way of reasoning is explained in more detail in [2].

Agents make local decisions that do not require direct communication with other agents. We try to exploit localised knowledge and criteria, and highly summarised information about the environment. Agents also act autonomously. They control and are responsible for their resources. While trying to maximise their wealth, they put a position according to what they have inferred from the signals.

A basic accounting procedure takes place for every agent. At the beginning of each trading cycle, this process is executed by updating the trader's new holdings and cash accounts. First, the holdings account of each trader is updated by adding (if there are splits or the trader is buying shares) or removing (if it is selling) the number of shares involved in the transaction.

The same principle applies for the cash account. Cash is updated by adding (if there is a dividend payment or if the trader is selling) or subtracting (if the trader is buying) the cash involved in the transaction. At this time a commission is also discounted for all transactions executed. If the trader decided to buy he must own all the cash incurred in the transactions, including the commission charged. If the decision is to sell, the trader must own at least the same number of shares he wants to sell.

Therefore in this market scenario there is no possibility to borrow money or to sell short. By the end of the day, the wealth of each trader is updated by adding the interest paid during one cycle to the cash account.

The wealth $w_{i(t)}$ at time t of agent i is given by the equation:

$$w_i(t) = (1 + r)M_i(t) + h_i(t)p(t) \tag{1}$$

where $(1 + r)M_i(t)$ is the cash invested at an interest rate r and $h_i(t)p(t)$ are the holdings calculated at the current price of the stock $p(t)$.

3 THE SIMULATION

There is a commission C of 0.1% incurred for each transaction. The interest rate paid by the bank has been set to 8% annually. The three agents are able to buy, sell or hold their possessions as they wish. On day one they are given the amount of $10,000 (ten thousand U.S. dollars) to invest. In addition to the initial cash, they can also be given a number of shares to start with. Any combination desired is possible.

Two non-intelligent agents have been introduced in the market for performance comparisons. They are allowed to trade only on day one, and are given exactly the same initial amount in cash and shares that is given to the artificial traders. Their accounting is calculated on a daily basis as well. One is called bank, and as its name suggests, if it is given shares initially, it converts them to cash, paying the correspondent commission. If it only has cash, it leaves it to compound in the bank for the total length of the simulation. The other one does exactly the opposite. It buys shares on day one and does nothing thereafter. This second agent represents the buy-and-hold strategy.

The idea is that the apportionment of credit algorithm, along with the genetic algorithm should provide each individual agent with experience-based rules for buying, selling and holding. We want to test whether this holds for the proposed model.

3.1 THE ADAPTIVE PHASE

In this first phase the simulation starts by randomly initialising the strategies of each agent, all with strengths values equal to 10. There are 100 classifier rules per agent, although they are designed to have any number of classifiers, completely independent of each other. The probability of having # symbols is 0.5. The rest is divided equally to fill with 0's and 1's. The reason we set the probabilities this way is because we want to start with more general rules to match the given environmental states. The action part has no # symbols, so in this case there is equal probability of 0.5 to fill the four action bits.

The measure of performance of any given agent is the amount of money it accumulates through its actions and it is calculated at the end of each day.

The full set of data we used for the two sets of experiments corresponds to 10 years of real prices and volumes for the stock Merck & Co. Splits and dividends declared by the Company over that period are already discounted in the price series, so there are no jumps in prices due to splits or in wealth due to payment of dividends. For the first set of experiments, the random scenario, we only used the first 9 years of data. This set of data is taken as a training set, while the remaining year will be used later on for testing purposes.

Nine years of financial data briefly correspond to 2,275 trading days. We ran several simulations with different seed values with three agents trading daily to select the best performance of each type of trader recording the parameters used as well as the rule configurations evolved.

The parameters for Trader Type 1 are shown on table 4 along with the parameters for the other two traders. Automatch flag refers to a covering algorithm we designed similar to the one suggested in [7], to create a matching classifier when there is no current match and insert it in the population replacing the classifier with the lowest performance. The action is selected according to price behaviour (i.e if the price today is higher than yesterday, the action is set to 1) and the proportion is selected randomly. The classifier's strength is chosen to be the maximum value between 10 and the average value of the population's strength. Additional experiments showed us that inserting the initial value of 10 in all cases did not improve performance because in several occasions, a non-matching situation was encountered when the average of strengths was well above 10 and the inserted classifier became one with very low strength and ended up disappearing, either because other non-matching classifier was inserted, or because the genetic algorithm would select it for replacement, forcing the system to make unnecessary re-insertions of the same classifier later on when the same environmental message appeared again. In all runs with Merck & Co. data the use of the automatch algorithm showed improvements.

GA period refers to the number of generations or cycles that have to pass before the genetic action takes place. In all cases presented in this paper, when the GA is ON, 20% of the current population undergoes reproduction, crossover, mutation and replacement by crowding. The following parameters are straight forward from [4]: specificity, noise, and bucket brigade flag, along with others not shown here such as life tax, bid, ebid, etc.

Table 4: Parameters for Adaptive Simulation

Parameter	Tt1	Tt2	Tt3
GA period	200	100	100
Automatch Flag	ON	ON	ON
Specificity Flag	ON	ON	ON
Noise Flag	OFF	OFF	OFF
Bucket Brigade Flag	ON	ON	ON
Reinforcement	0.5	0.5	0.5
Number of shares	0	0	0
Initial Cash given	10,000	10,000	10,000
Commission per trade	0.1%	0.1%	0.1%
Annual Interest Rate	8%	8%	8%
Seed	0.9	0.95	0.9

Figure 1 shows the wealth of Trader Type 1, the bank (static investment at a fixed compounded interest rate) and buy-and-hold strategy over the 9-year period. Even though the agent starts with randomly selected classifiers, it manages to outperform both of the strategies shown. As it can be seen, at around day 750 the agent started to outperform the buy-and-hold strategy by far. It is at this point when the agent must had found and used some rules that proved to be profitable such as *buy when the moving average of the last 6 weeks is lower than the moving average of the last week* and so forth.

Figure 2 shows the number of shares the agent owns during the same period of time. The biggest oscillations in shares happened around day 1,000, exactly when its wealth outperformed the buy and hold strategy.

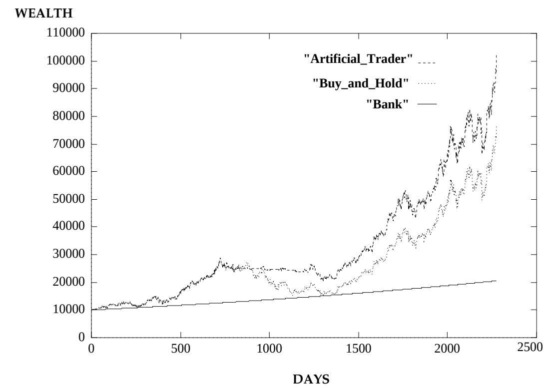

Figure 1: Wealth (in US dollars) of Bank, Buy and Hold Strategy and Trader Type 1

These particular results of wealth correspond to a single run. However, runs using different random number seeds show considerable differences in performance.

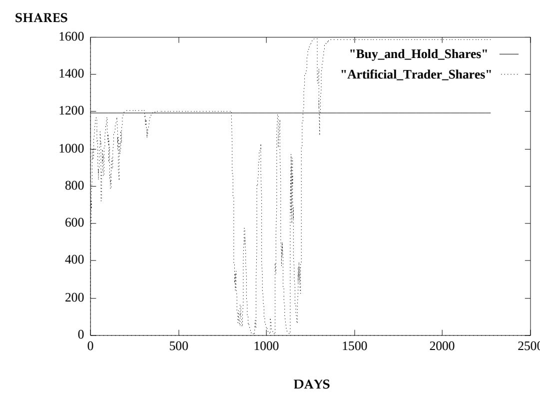

Figure 2: Number of shares owned by Trader Type 1

Figure 3 shows the distribution of shares for Trader Type 3 during the nine-year period. This trader also shows an increase in transactions around day 1,000, but in addition to this, there is another period of oscillations around day 2,000. It starts buying and selling in higher proportions which result in an increase of its wealth.

A specific examination of when rules fire showed that agents do indeed learn to buy on the upward trend and sell on the downward trend, although the precise details vary from agent to agent. In particular, space does not allow us to discuss what each agent regards as a trend.

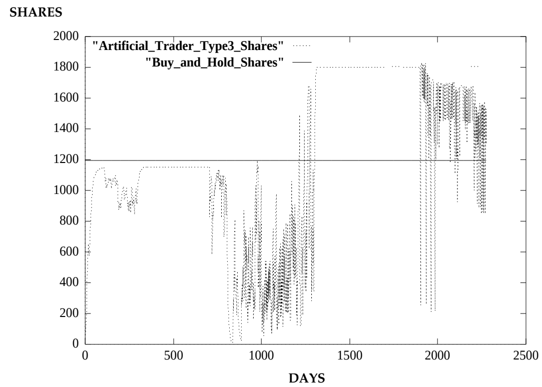

Figure 3: Number of shares owned by Trader Type 3

Another important consideration in the model is that the actions stipulated by the agents have been scaled. We introduced a "scaling" factor in the proportion of the action so the earlier in the run, the less effect their actions have. The total number of days was divided in 10 segments, the first one representing only 10% of the desired proportion to trade, gradually increasing each segment by 10%. So if on day 1 the trader wanted to buy 100% of its money in shares, a transaction of only 10% of that was actually performed. By day 2,000 it is allowed to trade as much as it wishes. Without this scaling factor, a trader could be wiped out by either bad decisions or bad price movements before it got the chance to learn about trends. This is the reason why figure 3 shows smaller number of shares traded at the beginning, i.e., there are no purchases of the same amount of shares as the buy-and-hold strategy (1,200 shares). But these oscillations gradually increase until around day 2,000, when the swings of share transactions ranges from 200 to 1,800 in only one trading cycle.

Table 5 shows the initial and final wealth of the bank investment, buy-and-hold strategy, Trader Type 1, Trader Type 2 and Trader Type 3 in runs with the parameters previously described.

3.2 THE NON-ADAPTIVE PHASE

The purpose of this phase is to test whether the credit assignment and the genetic algorithm from the experiments of the previous phase managed to evolve agents that can understand the relevant phenomena well enough to make useful adjustments in future - unseen- events and act accordingly. It is non-adaptive only in the sense that there is no genetic action involved.

All classifiers start with strength equal to 10. This simulation runs for a total of 253 trading days, equivalent to one calendar year, starting the day immediately after the previous phase ended. Refer to table 6 for parameters different from those shown on table 4.

Table 6: Parameters for Non-Adaptive Simulation

Parameter	Tt1	Tt2	Tt3
GA period	OFF	OFF	OFF
Automatch Flag	on/off	on/off	on/off
Seed	N/A	N/A	N/A

Figure 4 shows the wealth of Trader Type 1 against the wealth of the bank and buy-and hold strategies, followed by the shares distribution on figure 5. We have drawn a square around day number 2,425 in both graphs to show that the kind of decisions taken by the agent at that specific time contributed to its accumulation of wealth. In the square of figure 5 there are three drops in shares that correspond to selling at peak or near peak prices. Results for the other two types of traders are consistent with the ones shown here.

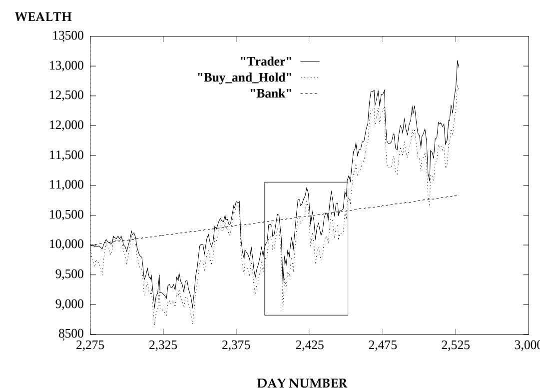

Figure 4: Wealth (in US dollars) of Bank, Buy and Hold Strategy and Trader Type 1 over Test Set

Summarising the results of this non-adaptive phase, table 7 shows the initial and final wealth of the bank investment, buy-and-hold strategy and Trader Type 1, for runs with the parameters previously shown.

3.3 HOW MUCH FURTHER? ...

Let's run an adaptive simulation over the already evolved set of classifiers from section 3.1 but this time using the testing set of data from section 3.2 to see if we find something interesting. On table 8 we report results using the GA every 20, 50, 80 and 90 out of the 253 days. The last column refers to results previously shown with the GA turned off. These figures reflect

Table 5: Returns of all Agents from Adaptive Simulation

2,275 trading days	Bank	B&H	Tt1	Tt2	Tt3
Initial Capital	10,000	10,000	10,000	10,000	10,000
Final Capital	20,546	76,733	102,180	82,006	100,755
% Increase with respect to Bank	0	273	397	299	390
% Increase with respect to buy-and-hold	-73	0	33	7	31
% RETURN with respect to initial investment	105	667	922	720	908

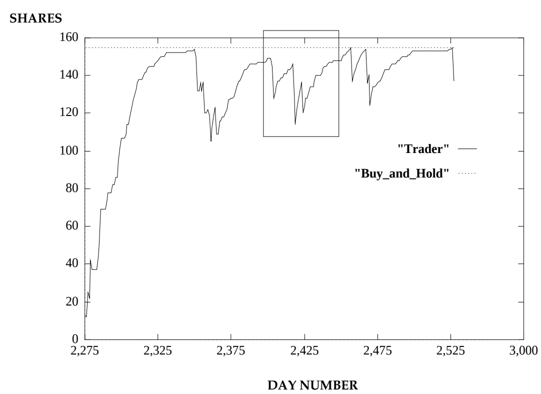

Figure 5: Number of shares owned by Trader Type 1 over Test Set

Table 7: Returns of Tt1 from Non-Adaptive Simulation

253 trading days	Bank	B&H	Trader
Initial Capital	10,000	10,000	10,000
Final Capital	10,834	12,587	12,968
% Increase with respect to Bank	0	16	20
% Increase with respect to buy-and-hold	-14	0	3
% RETURN with respect to initial investment	8.34	25.87	29.68

only the **averages over 11 runs** with different seed values in the range of 0.0 to 1.0, increasing in steps of 0.1. Again, space did not allow us to show single seed results. Although results appear more detrimental because "on average" performance drops when adding further learning to the set of rules already learned, taking a closer look at the individual performances with each seed, we were able to find the highest performing individual (with GA every 50 generations) found so far over this data set, which represents a 23% increase over bank, and 6% increase over buy-and hold strategy.

Table 8: Returns of Tt1 from Adaptive-Adapted Simulation

253 trading days	GA 20	GA 50	GA 90	GA OFF
% Increase with respect to Bank	8	17	19	20
% Increase with respect to buy-and-hold	-7	1	3	3

3.4 ADAPTIVE VS NON-ADAPTIVE

In order to analyse separately the learning due to apportionment of credit among the original (random start) rules and that due to the injection of new rules by the GA, on table 9 we show comparisons in average performances of runs with various values of GA invocation (GA ON) against runs in complete absence of genetic action (GA OFF). In these experiments we used the original 9-year set of data, starting with 100 random classifiers, which should be able to contain a moderate subset of rules (although not very rich for this difficult problem). The values represent the % of increase in wealth with respect to the bank.

Table 9: Testing the relevance of the GA as a rule discovery heuristic

GA period	% Wealth Increase
50	163
100	155
200	129
500	125
1000	90
OFF	113

We will consider the case of GA ON every 100 periods to give an idea of the effect this can have over the population of rules. 20% of the current population of 100 classifiers undergoes reproduction, crossover, mutation and replacement by crowding, meaning that the population of rules with a GA invoked every 100 cycles is turned over 4.55 times in 2,275 iterations.[1]

Results with the GA OFF indicate how the apportionment of credit algorithm adjusts the strength values, and those with GA ON indicate the effect of evolving new rules, which indeed is increasingly beneficial, except in the case in which the GA is invoked every 1,000 iterations. This means that it is only called 2 times during the total number of days, so when it is called, it only does a little damage to the rules. All the other cases show an improvement in average performance.

4 CONCLUSIONS AND FUTURE WORK

In this model we have begun to address the problem of developing multi-agent strategies for trading successfully with minimal – and different – domain knowledge and information sharing between agents. In this simple multi-agent setting, we have shown how all three types of traders were able to survive and furthermore, outperformed the bank and buy-and-hold strategies in both sets of experiments.

Learning Classifier Systems seem to be able to at least represent competent traders. The agents were successful in finding profitable rules exploiting the market inertia. For example, making money by selling in a bull market until the trend changes, when they start buying again. Therefore technical trading is a valid outcome in this market.

It is also interesting to note that although each agent is independently using reinforcement learning techniques to optimise his own environmental reward, there is a global incentive to trade without having any explicit or implicit information sharing between the agents.

For this market model to be truly artificial, the prices must be determined endogenously according to supply and demand rules. We are currently implementing this phase.

The fact that in this model each type of agent stands for the aggregate effect of a proportion of traders of its type simplifies the processing task enormously. Approximately, the run time of a single simulation described in this paper with 9 years of historical data of Merck & Co. takes about 90 seconds on a SUN Ultra 5 working station. Therefore it would be feasible to add more features to this single-market/single-good economy such as having more instruments to trade or even more markets interacting.

[1] (0.2)(100)(2,275)/100 = 455 new offspring

Acknowledgments

The first author wishes to thank Consejo Nacional de Ciencia y Tecnología and Fundación Mexicana para la Educación, la Tecnología y la Ciencia, A.C., for their financial support.

References

[1] W. B. Arthur, J. H. Holland, B. LeBaron, R. Palmer, and P. Tayler. Asset pricing under endogenous expectations in an artificial stock market. Working paper, Santa Fe Instituite, December 1996.

[2] W. Brian Arthur. On lerning and adaptation in the economy. Working paper 92-07-038, Santa Fe Institute, 1992.

[3] Guido J. Deboeck. *Trading on the Edge. Neural, Genetic and Fuzzy Systems for Chaotic Financial Markets.* John Willey & Sons, Inc., New York, NY, 1st edition, 1994.

[4] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison Wesley, Reading, MA., 1989.

[5] John H. Holland. *Adaptation in Natural and Artificial Systems.* MIT Press, 1992.

[6] John H. Holland. *HIDDEN ORDER. How Adaptation Builds Complexity.* Addison Wesley, 1995.

[7] S. W. Wilson and D. E. Goldberg. A critical review of classifier systems. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 244–255. Morgan Kauffmann, 1989.