A Learning Evolutionary Trading System LETS

Sonia Schulenburg and Peter Ross
s.schulenburg@napier.ac.uk, peter@dcs.napier.ac.uk

c/o Evolutionary Computation Research Group
School of Computing,
Napier University
10 Colinton Road,
Edinburgh EH10 5DT

1 Forecasting Financial Markets

Many techniques have been used to model financial time series data such as stock prices, including linear and nonlinear statistical methods and more inscrutable methods such as neural nets. The hope is that if the time series can be predicted fairly well, the predictions can be used to make profitable buy/sell/hold decisions. Time series modelling may work well for physical systems, such as blood flow in veins, because the underlying physical laws do not change with time even though their effects (such as the rhythmic expansion and contraction of blood vessels) may do so. However, time series modelling in the financial world often seems to fail to track any of the causes of the data, and the causes themselves change over time. All that happens is that the model ceases to fit the real data well, and then it has become time to re-train or re-fit.

So what can we do? Continual re-training may not be the answer; in the case of (e.g.) neural networks it may be expensive, and certainly performance cannot always keep improving, so there is a potential problem about when to switch from the old model to a new one. Another drawback of such models is that they do not tend to identify or explain anything about the relevant phenomena that is being modelled; this is not their main goal. As a result of such poor explanatory capabilities, there is no added knowledge or hypothesis built about the underlying causes of the changes in the environment being modelled.

The approach here is different. Because financial markets are complex systems where the players are human traders, the aim is to address these issues by focusing on the agent rather than on the data. The idea is to model traders' decision-making processes, as they decide repeatedly whether to buy, sell or hold a particular stock.

Modelling individual traders may not be ideal, individuals are extremely complex, their actions are affected by their private knowledge, their emotional make-up, sometimes even their breakfast! Why then, not model the behaviour of whole groups of traders instead? The underlying hypothesis is that the aggregate behaviour of a group will be simpler to model than that of individuals and that it may still produce very good results.

Modelling the behaviour of a *group of traders* seems to be a sensible idea. Why? In this work each group of traders uses a certain set of real market information before making decisions. To make a simple start, the groups receive small and reasonable-looking sets, consisting of some binary indicators about how today's prices and volumes relate to various moving averages and historical extremes. The behaviour of the group is expressed as a set of rules that relate to a single stock, such as: if A and B and not C and D then buy 45% of what cash-in-hand allows; if not A and B and not C then sell 15% of current holdings; etc.

It is important to emphasise that *trader types* are evolved rather than models of individual traders. The assumption is that the aggregate behaviour of a set of traders, each of whom is (say) basing buy/sell/hold decisions on certain sorts of data such as information about various moving averages and recent highs and lows, will be simpler to describe than the complex behaviour of only one trader. Any one trader might include in his decision-making factors such as personal knowledge of the people controlling the company, private knowledge of the local economy, etc. However, within a group who use the same basic methodology, the effects of such extra factors may tend to balance out so that the group's behaviour may at least be approximated in terms of only a modest set of easily available indicators.

For example, a *trader type* may pay attention to such factors as whether today's price is more than 20% higher than the 5-day-moving average, whether today's volume is larger than yesterday's and so on. At present, a *trader type* is characterised by the particular set of such factors f_i – each of them binary, a simple yes/no – that it pays attention to. Its behaviour is expressed in terms of rules whose conditions are a combination of terms such as $f_1 = \text{yes}$, $f_2 = \text{no}$, $f_3 = \text{don't care}$ (the *don't care* are wild-cards, they match either yes or no). Rule actions can be to do nothing (hold), or to buy more stock using a certain proportion of current cash invested in the risk-free bond, or to sell a certain proportion of holdings in the stock and invest that in the risk-free bond.

The goal of each of these *trader types* is simple. They are all trying to learn behaviours (rules, market strategies which they create and improve) that will lead them to increased profits under the current market conditions; they are not looking for the *optimal* behaviour, only an appropriate one. The system does not directly predict price values (e.g., it does not tell you that the price tomorrow will be £123.86 like some NN models try to do) or the direction of change (price will rise or fall), but rather it delivers a specific action (chosen from a number of competing strategies) for the agent to follow, and such action is be rewarded only if it turns out to be profitable.

2 The Model

In summary, the goal here is to create a number of *adaptive agent-types*, capable of analysing and classifying various historical and non-historical market data to incorporate in an investment decision-making process where the agent will trade upon in order to survive and hopefully grow in assets. The main concept is that the *agent-type* is born with no prior knowledge about the market and it is forced to start trading its funds (buying, selling or holding) since day one. The simplistic on-line nature of this learning process is one of the main features of the model. The decision-making process can be performed daily or intra-daily, depending on the availability of the data. This will be explained in more detail later in this section.

This section briefly describes the structure of the model, which can also be found in our previous papers [Schulenburg & Ross 99, Schulenburg & Ross 00, Schulenburg & Ross 01, Schulenburg & Ross 02]. The market structure consists of the following elements, along with the roles they play in the trading process:

1. **Time**, which is discrete and indexed by *T*, represents one cycle equivalent to one *real* trading day in the market. There are only about 251 trading days in a year due to weekends and holidays, so if we refer to a ten year period of historical data, it roughly corresponds to a total of 2,510 days.

2. Any **real risky stock** *S* being traded in any stock exchange. This corresponds to real data; examples are Microsoft Corporation (msft), Forest Oil Corporation (fst), etc. The stock to analyse *S*, as well as the length of time series *T*, are selected by the user.

3. The **buy-and-hold** agent, which represents the *buy-and-hold* strategy. This agent is given an initial amount of money – in the currency of the stock being traded – and converts it into the stock, keeping the stock during *T* time periods. The buying price $P(T)_0$ corresponds to the stock's price of the first time-step. There is a commission *C = 0.1%* (set by the user and can be a flat rate such as $10 USD per trade or any other) charged when doing this single transaction.

4. The **bank** agent, which keeps all money in the bank at a good rate of interest *R*, never buying the stock. Therefore this agent does not own any shares, all its possessions are cash, compounded in the bank at an interest rate such as *R = 8%* p.a. (also set by the user and can be variable). When given shares, it immediately sells them, paying the appropriate commission for the transaction.

5. One **trend-following agent**, representing a strategy that varies according to price moves. This is a type of *momentum trader*. The *trend-following* strategy is simple and yet will outperform both the *buy-and-hold* strategy and the *bank* strategy if the stock price does not show a clear long-term trend upwards or downwards. It assumes that there will be an uptrend if the current price of the stock is higher than last period's, and therefore it buys the total number of shares that its available cash allows, minus the commission. If the current price is lower than the previous price, it assumes there will be a continuation of that downward trend

and sells all shares in possession. In both cases no other transaction is made until there is a reversal in trend. Note that when this strategy sells its stock and puts its money in the bank, the commission it pays on the transaction is higher than three days of interest earned on the bank deposit[1]. On any given time-step, its holdings are either all shares valued at the current price, or all cash in the bank earning interest.

6. One **random walk agent**, who makes a daily random decision of whether to buy, sell or hold the stock. This agent pays the same commission set by the user and it is relevant for two reasons: first, because we are interested in comparing the performance of our *adaptive traders* against a decision that does not involve any learning at all; and second, it is usually derived from traditional economic theories such as the Efficient Market Hypothesis (EMH) and the Random Walk Hypothesis (RWH), that on the basis of past price patterns, investors can not develop a strategy that yields abnormal profits [Malkiel 99]. Therefore many papers report results of decision-making systems of various kinds against this strategy, with the view that outperforming it – even by a small percent – is difficult to achieve and shows that such models are producing sensible results that could lead to abnormal profits. Similarly, predictive capabilities of linear and non-linear methods are usually compared against the *random walk strategy*. It is important to note that it is not our intention to enter into the long debate of whether prices do or do not follow a random walk ([Malkiel 99] vs. [Lo & MacKinlay 99]), what matters here is to check whether our *artificial traders* achieve superior *profits*[2]. (By simply comparing how much profit the *random agent* makes, against the *adaptive traders* profit's.) Note that commissions are still being charged for every transaction.

7. 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.

   For instance, the following represents a typical format in which most financial data is freely obtained through the Internet. This portion of data was taken from `http://quote.yahoo.com` and it corresponds to the stock of Coca Cola during the month of April, 2001. It looks as follows:

```
Date,Open,High,Low,Close,Volume
30-Apr-01,46.75,46.75,45.85,46.19,3837000
27-Apr-01,47.50,47.50,46,47,3480900
```

[1] Trading days in a calendar year = 251. Commission charged for selling \$10,000 US Dollars of shares is $C = \frac{0.1}{100}(10,000) = \$10$; 3 day's Interest payment $R_3 = \frac{8/100}{251}(3)(10,000) = \$9.6$.

[2] For a different approach which formalises the notion of *profitability* and *predictability* in regards to the EMH, refer to [Chen & Yeh 96a, Chen & Yeh 96b], where it is shown that a Genetic Programming (GP) model outperforms the random walk, confirming the belief that certain non-linear regularities exist. However, the argument is that EMH is sustained because the search costs involved in discovering them might be too high to make their exploitation profitable.

```
26-Apr-01,47.60,47.98,46.90,46.90,4296400
25-Apr-01,47.50,48.40,47.40,48.20,3247700
.
.
.
2-Apr-01,45.40,46.92,44.86,45.85,5212400
```

As it can be seen, the series starts with the `date`. Date formats vary between sources. The following four columns describe the various prices that are recorded daily, starting from the `opening price` of the day, then the `highest` and `lowest` of the day, followed by the `closing price`. Finally, the last column displays the `volume` of transactions relative to the stock. Note that dates are given in descending order, so they should be re-ordered in ascending order before being processed, this is a simple step which can be done by using almost any spreadsheet package.

This raw information is then processed by the *trader-types* in different ways. For instance, an agent might take into consideration factors such as the first difference in price, or a certain moving average of the stock's price or volume. While choosing the right information set, it is important to explore whether a *trader type* could benefit from more, or maybe less, information. Using more factors provides more clues but also multiplies the size of the search space so that the evolutionary process may come to be governed more by neutral genetic drift than by genuine selective pressure. Other sorts of information, such as external political or economic conditions, might be introduced in a simplistic way by, say, adding the behaviour of the FTSE 100 index (or similar gross indicators) as an extra factor.

At this point, another aspect to mention is that the stock prices used are *daily closing prices*, which are not necessarily the *true prices* at which one could buy it at present or in the near future. At any given instant in time, there is a best or highest "bid" price from a potential buyer and in the same way there is a best or lowest "ask" price from someone who wants to sell the stock. Using closing prices is less than ideal, since nobody can actually trade at those prices, but bid/ask prices are not easily (nor freely) obtainable. In addition, daily data restricts the model to making only daily decisions. However, more frequent trading decision points could be achieved through the use of high frequency data. This would allow the system, rather than trading at most once per day, and using only closing prices, use continual information and either have several decision points per day, or make the decisions event-driven.

8. Any number of ***trader-types***, and any number of traders per type, which are heterogeneous agents designed to learn and adapt to a market environment that is partially understood and where the domain characteristics can change rapidly over time.

   In order to keep the model as simple as possible, only up to three types of traders have been introduced so far, but the user can easily alter this and build as many as he/she wishes, as long as there is available market data to build them, i.e. one type could be more technical, another more fundamental, etc.

Figure 1 shows that the system can have *N* number of *trader-types* and each one of these types can have *M* individual traders. Each one of these have *R* number of market strategies. *R*, *N* and *M* can be set by the user. Previously we have fixed *R* to 100 rules per trader, but our new model of LETS this can vary from trader to trader. For instance, in our previous papers we have shown three *trader-types* – note that all receive different sets of market information, and 1,001 traders of each type, which is equivalent to the number of runs we perform. In this paper we only use one type, which comprises most of the information the other three had, and 1,000 traders of this type.

As seen in the figure, each trader is fed with the current market information (inputs); the output is a given action (buy/sell/hold), which is then performed and the environment feeds back a given reward if the action turned out to be profitable. Rules keep improving through a reinforcement mechanism and new ones are created and injected through the use of a Genetic Algorithm (GA).

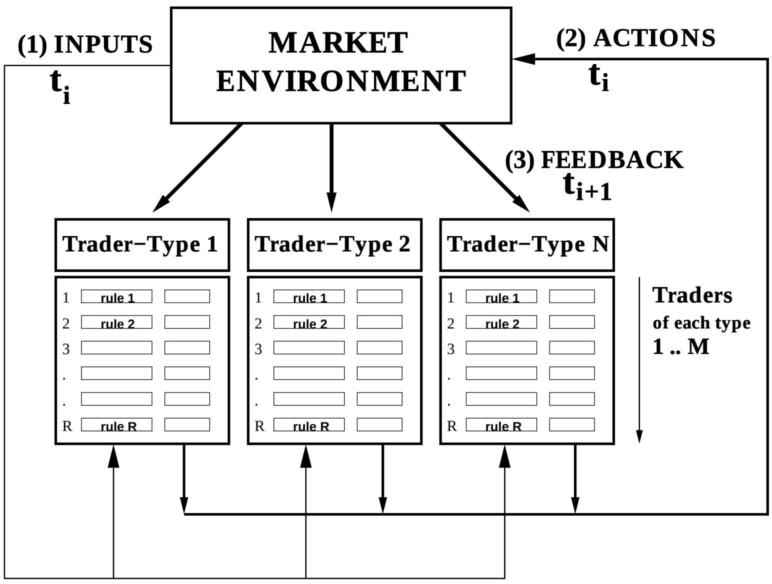

Figure 1: Structure of *Trader-Types* and the Relationship with their Market Environment.

One type of agent can not evolve into another one, but it can go broke and thus halt while another one can get rich. It is also possible for an agent to learn to ignore any particular field in the daily market information it receives, but it can not ask for extra fields beyond those given to it. Agents are not allowed to borrow money if they run out of it.

Note that the stock price does not change according to the supply and demand governed by the artificial *trader-types*, but rather by changes of real phenomena outside their scope. We are dealing with **real data** here, this is not a truly artificial stock market. The assumption is that small transactions would not have an effect in the market as a whole; usually only $10,000 US dollars are given to the agents to trade.

9. An **accounting procedure**. This procedure simply calculates and updates the possessions of every *trader-type* and strategy, according to the new price. Possessions at any given time include the shares owned by each *trader-type*, their current holdings, interest payments, cash available, and their current wealth. The procedure works as follows: at the beginning of each trading cycle, the agent's holdings and cash accounts are updated, including stock splits, cash dividends and commissions for all transactions executed. For instance, if the trader decides to buy, it must own all the cash incurred in the transactions, including the commission fee. When selling, the trader must own all the shares it wants to sell. Agents cannot borrow money or sell short. At 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 thus 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 Learning Process

Continual/Non-repetitive Learning is a novel property of this model. In simple terms, it means that the stock price on any given day of the simulation is never seen more than once, and that the learning process never stops. The agent is trained according to how successful it is in the real market where it is trading. The process differs from other time series analysis models such as NN and GP in at least two ways: first, in these models the data is divided in separate training and testing sets. Training is usually performed using a large portion of the available data e.g. the first 90% and testing using the remaining 10%. Typically, when training is complete, performance is then tested on the unseen data and the results obtained are reported separately as well (in-sample versus out-of-sample performance).

In [Schulenburg & Ross 99] results with separate sets of training and testing were reported. In these experiments, training was performed using 9 years of data and testing using the last year, during which the GA was turned off. In that particular example (Merck & Co.), the testing phase did produce better results than the *buy-and-hold* and the *bank* investments. This is useful to demonstrate that something useful has been learned. However, we believe that it is unrealistic to switch off training in practice; in practice, traders do not use a fixed set of rules over an indefinitely long period of time but change them as market conditions differ. Therefore the adaptive learning should not be frozen in non-stable environments such as real stock markets.

Secondly, if using a NN or GP instead, data may be presented for training e.g. through back propagation algorithm, literally thousands of times as part of the process of trying to minimise an error measure such as the Mean Squared Error (MSE) (while trying not to increase it in the test set). Because of this training regime involves a large number of updates, such systems are usually very slow, needing retraining at uncertain intervals and can be unsatisfactory because they offer no convenient explanation of why a given buy/sell/hold decision was made.

In our model there are no separate training and testing phases. All the data are unseen, and the learning process is a continual one. The test of success is whether an agent continues to trade profitably, especially when compared with plausible non-evolutionary strategies. It is not in the interest of this work to try to examine whether it is possible for an artificial stock-market agent to be trained to learn good trading behaviour through repeated encounters with historical data, the goal is to explore whether such agents can survive in the most human-like way found so far: where opportunities are given only once and where market conditions are not guaranteed to be repeated.

Therefore we believe that the learning process should not be frozen in time, market behaviour is continually changing and learning and forecasting should be continual activities. Brian Arthur [Arthur 92] in the Santa Fe Institute (SFI) stock market has tested this view, by injecting into his populations of strategies some that had been very good in the past. He observed that such transplanted strategies behaved badly in their new market environment; clearly they had been adapted to specific past conditions. As market behaviour remains unstable and never settles down, learning and forecasting should be continual activities. Market strategies should genuinely adapt *continuously* because market conditions are ever-changing.

Chen and Yeh also formulated Arthur's survival test in their model by finding that the number of traders with successful searches starts high and then decreases steadily up to a certain point. These findings suggest that initially, traders find a secret but as they also change market dynamics by bringing the knowledge from the *school* of strategies back into the market, then more and more find it, making it no longer a secret. They argue that new patterns are created while exploiting the current ones, creating a "*self-destruction-and-organization-process*" [Chen & Yeh 99].

Another piece of evidence suggesting this never-ending process of market evolution was found in [Beltrametti *et al.* 97], where the evolved past strategies are no longer good in the future. Luca Beltrametti's learning to forecast the foreign exchange market experiment with an adaptive agent, shows that the out-of-sample forecasting ability of the adaptive agent under performed while in-sample forecasting outperformed the performance of forecasts given by Vector Auto Regression model (VAR) estimations of the exchange-rate's determinants. It is important to stress that the authors' purpose was to use the other methods as control devices to test the adaptive agent's goodness of fit by means of a formal statistical tool, i.e. whether the agent could learn to forecast the exchange rate under the conditions they specified in the experiment, not to compare if the adaptive agent was better or worse than the other models. For these reasons, it is suggested here that training and testing phases should not be performed separately, the model must function on-line and new strategies should be evolved fast, easily and cheaply.

4 Some Questions About LETS

This section briefly summarises some answers to typical questions regarding the traders, environment, reinforcement and discovery components of the model.

1. What do these *trader-types* know? Initially, nothing. The system starts with random strategies, but as time progresses, they develop through experience and discovery, better sets of strategies to trade upon. By looking at the evolved rule-sets, it is clear that the initial random strategies have not survived long; they are quickly improved or replaced by better ones because of the constant changes in the market environment and the learning process involved.

2. Can agents start with prior market knowledge? Yes. This can be done in several ways:

   - Agents of the same type, as well as agents of different types, can share their *actions*. We have experimented with this issue. The previous action of *trader-type 1* (Tt1) was given to *trader-type 2* (Tt2) as part of its information set. Usually, when Tt2 was not as good performer as Tt1 – either because the market information given to it was not sufficient or because it failed to discover good strategies – it gradually learned to follow the actions of the better trader, improving its own performance.
   - Traders of the same type could share their *experiences*, i.e. their most profitable strategies. Perhaps performance could increase substantially because the agents would be acting in a cooperative manner.
   - Also, other types of interactions that can be analysed are through agents dealing with *different stocks*, i.e. traders of the same type but dealing with different stocks that present similar market behaviour, such as PepsiCo and Coca-Cola, Sony and Hitachi, etc.
   - Obtaining actual market expertise can be difficult and costly. For this reason the model starts with no previous knowledge. However, if *expert knowledge* was available, one would only need to add it in the initial rule-set rather than having a purely random start and let the system run as usual.

3. What do these *trader-types* believe? One could argue that they believe in the EMH. In Learning Classifier Systems (LCS) language, this is a single-step environment. There is no link in rewards given for good actions for more than one step ahead. In other words, the environment does not reward chains as it is the assumption of this model that prices follow a random walk. A good action taken in the current time-step is rewarded in the following time-step only if it was successful at meeting certain criteria imposed by the designer. However, we plan to implement the multi-step environment where the agent will receive delayed rewards, i.e. if after one month its wealth is higher than the *buy-and-hold*, etc.

4. What do they learn? The agents are capable of adapting well to many very different kinds of stock behaviour. Here only the *stock market* scenario was addressed, but there are a number of other possible learning environments where this model

could be used. For example, it is well suited for real world applications where historical data in the form of time series is available such as in the insurance market, credit rating and fraud detection, foreign exchange markets, etc. Other applications where it can have good potential are in the development of systems for intelligent homes and to design better transport systems, to name a few.

5. How fast do they learn? The learning parameter is controlled by the designer, and which exactly is the best value depends on the stock being analysed. For certain stocks, faster learning rates are proven to be better than slower ones. Injecting more strategies too quickly does not give the agent a chance to test these models before they are replaced by new ones.

6. What exactly is the forecasting model the agent uses? Agents try to maximise their profits by following a process of adaptation. Agents working on the same stock usually differ in their pool of strategies, even if they are of the same type. This difference can be explained because as they learn through experience, their learning experiences can follow very different paths from one another; this property is common in human traders, which is exactly the target to model. However, consistency has been found in the evolved pool of strategies, in the sense that rule-sets cluster. That is, two evolved rule-sets for a given stock are more similar than two rule-sets from two different stocks.

7. How sophisticated are the traders? Again, as with the learning parameter, it is up to the designer to define how sophisticated to build these agents. For example, an agent with a condition length as long as 70-80 bits can be made (such as the first versions of the SFI artificial stock market model [Palmer *et al.* 94]). For the type of learning (with real data) we perform here and the short period of time we use, we recommend shorter lengths, e.g. 10 bits long or less for 3,000 trading cycles. Another type of sophistication can be measured through the *specificity*[3] parameter associated to every rule, i.e. the more specific and accurate the rules are, the better trading models are developed.

   Determining which items of information are more relevant than others or whether they are important at all can be achieved by looking at what the overall effect of subtracting these bits of information could have on performance. For example, there is considerable scope for experimenting with the mixture of rule conditions as a way of assessing whether we could manage to improve performance even further, which is necessary in order to guide us in the design of better trader models.

8. Not all runs produce *trader types* that perform well. How much effort does it take to produce a good agent through repeated runs? Because there is no repeated-training involved, i.e. the data is only seen once, these models can be evolved very quickly, taking only a few seconds to run the system from a pure random start, with, 10 or more years of daily data. It has been shown that in all

[3]The *specificity* S of a given rule determines the number of non # symbols in its condition part. The S value of classifier `01110101:100` is 8; of `#1#0#10#:100` is 4.

cases where 1000 runs were made, a reasonable number of models were found to consistently beat the *buy-and-hold* strategy. This number of winners is, of course, stock dependent. In stocks like Forest Oil it is much easier to beat the *buy-and-hold* strategy than other extremely good performers such as Microsoft.

9. Is the agent stock-dependent? Yes, agents are very much dependent upon the characteristics of the stock in question. The agent's rules contain two parts, the market condition and the action to follow. The condition matches a given environmental state, therefore the sets of evolved rules have the market characteristics embedded in their condition part, and associated to these is the action.

5 Experimental Results of LETS with Single Stocks

All experiments shown here use daily data, so that the trader's decision is performed on a daily basis, at the end of each day, and using the *close price* of the stock, along with other factors such as highs and lows and Moving Average (MA) comparisons with the current price. As we said before, performing a large number of runs is relatively easy, taking only seconds to do 1000 runs in a decent PC. For every experiment of 1000 runs (different seeds), we select the best performer and display the results obtained by this agent. Why is the best performer more important than, let's say, the average of the group?

Initially there is no knowledge injected in the system and the agents are forced to start trading from day 1. As they build their pool of market strategies, some of them might be more lucky at the beginning and some might lose more money before learning about the relationships between the data they analyse. Every trader of every type (in this case 1000 traders of one type) follows a different evolutionary path and some are more successful than others. We have experimented with this, by not allowing them to trade all they wish early on, when they have not learned much. However, we have found that the good traders keep improving quite steadily, and what started as good luck developed into a skill. Therefore we are interested in obtaining best performers only. This is a common sense choice. Most papers report only the best results out of a large number of runs, specially in NN models where there are so many design issues involved, such as the net architecture and learning parameters. The best performers out of the group would be the traders we would trust when making decisions, not the averages; there is no single trader associated with the average performance.

Table 1 shows the companies and the stock symbols we use in these experiments. They have been selected for no special reason – this is a very difficult period to analyse. It is clear that all stocks have been badly affected by the attacks on September 11th 2001, and even before the incident, the technology sector had already fallen dramatically. As it can be seen, most series are over 3,000 days long and they all end on May 15th 2002. However, not all start on the same date; a couple of them (Nokia Corporation and Lucent Technologies Inc.) started trading later on.

For every one of these stocks, Table 2 displays the final wealth of an initial investment of $10,000 US Dollars in the *Bank, Buy-And-Hold, Price-Trend, Average Random, Maximum Random*, and the *Artificial Trader*.

Table 1: Stocks Analysed

Stock	Company Name	Num-Days	From	To
csco	Cisco Systems Inc.	3,063	26Mar90	15May02
fst	Forest Oil Corp.	3,058	26Mar90	15May02
hit	Hitachi Ltd	3,121	02Jan90	15May02
ibm	IBM Corp.	3,121	02Jan90	15May02
ko	Coca-Cola Company	3,121	02Jan90	15May02
lu	Lucent Tech. Inc.	1,539	04Apr96	15May02
mrk	Merck & Co. Inc.	3,121	02Jan90	15May02
msft	Microsoft Corp.	3,121	02Jan90	15May02
nok	Nokia Corp.	1,778	25Apr95	15May02
pep	PepsiCo Inc.	3,121	02Jan90	15May02
sne	Sony Corp.	3,121	02Jan90	15May02

Table 2: Final Wealth in US Dollars. *Initial Wealth = 10,000; Commission = 0.1%*

Stock	Bank	B&H	P-Trend	Ave-Ran	Max-Ran	Trader
csco	26,367	2’067,914	19,442	154,790	1’005,395	6’890,896
fst	26,325	3,242	369	7,742	62,534	179,123
hit	26,856	7,991	13,551	7,792	24,598	66,737
ibm	26,856	34,784	9,659	16,383	101,365	136,149
ko	26,856	68,311	21,559	21,980	60,755	141,325
lu	16,271	6,610	35,605	9,151	51,419	189,137
mrk	26,856	56,700	15,709	19,997	72,555	114,903
msft	26,856	444,576	30,840	59,384	227,240	902,206
nok	17,556	64,060	59,428	28,190	170,061	346,639
pep	26,856	49,729	3,047	18,290	64,126	107,082
sne	26,856	22,530	24,175	13,034	48,890	133,510
AVE	24,956	256,950	21,217	23,339	171,722	837,064
% PROFIT	150	2,470	112	133	1,617	8,271

The average of the random walk strategy has been taken because of the high variance between different runs. However, the best of these 1000 runs is shown in column *Max-Ran*. Column *Trader* shows the wealth obtained by the *adaptive agent*. Included is also the average (*AVE*) of these investments and the *PROFIT* earned (as a percentage of the original investment). *Buy-and-hold* with final wealth of $256,950 (profit of 2,470%) is the best of the *non-adaptive* strategies, including the maximum of the random runs. However, the percentual gain of the *adaptive agent* (8,271%) against *buy-and-hold* is substantially higher – better by a multiplicative factor of 3.35. The *adaptive agent* has clearly outperformed every one of these strategies both, separately and on average. Note that the commission has been set to 0.1% for every transaction performed by all strategies shown, including the *artificial trader*.

6 Experimental Results of LETS as Portfolio Manager

The system has been extended to be used for portfolio management. The idea here is simple: each *trader-type* deals with one stock only. The initial cash e.g. $10,000 USD is assigned to *PM*, our portfolio management system's *arbitrator*, which decides which trader will be allowed to buy at any given moment in time. The funds that are not allocated in stocks are invested in the bank, as usual. Alternatively, these funds could be invested in a baseline stock deemed safe, playing the role of the bank.

First we created a portfolio composed of the 11 stocks shown in Table 1. Then the system apportions different amounts of money into the different stocks, depending on the trader's daily suggestions. For instance, if the trader of *msft* suggests a buy and *fst* a sell, then the system sells *fst* first and then buys all available cash into *msft* stock. If all 11 decide to buy on a given trading day, the system buys equal amounts of stocks from the available cash.

Table 3 shows the final wealth of the same strategies as shown in the previous section. The *Bank* works in the same way, investing the initial $10,000 at 8%. *Buy-and-hold* works slightly different: here the system invests equal amounts of the original $10,000 and waits until the end of the period. Note that the series are shorter, this is due to the fact that for multiple stocks, the system selects the shorter period, and in this case it corresponds to Lucent. *Price-Trend* works similarly, selling when price is decreasing and buying when price is increasing. In portfolio management there is no point in showing runs with the *Random- Walk* agent as we did previously; they perform poorly.

We show two types of rewards (a rule is rewarded if it holds the stock when price fluctuates within 2 or 5% of the previous price) and two types of commission. The first one is a fixed charge of 10 USD per trade for trades of less than 5,000 shares (and 0.1% for orders of more than 5,000 shares) and it is not permitted to buy less than 10 shares, against the typical 0.1% commission per trade we have used in the past.

The table also shows two versions of portfolio manager: *PM1* and *PM2*. In *PM1*, every agent starts with the same seed; here we do 5,000 runs, each with a different random seed and *PM1* is the best of these. In *PM2*, initial experiments (5,000 runs as well) are done first to find the best seed for each agent, and only one run of *PM2* is done at the end using those discovered seeds. We are currently experimenting on both approaches and some improvements are already on the way.

Table 4 shows the results with only nine of the stocks analysed in the previous case. We decided to take out the shorter series, Lucent and Nokia to allow the system to run for 3,058 days (the shortest period of the nine stocks) rather than only 1,539 trading days with 11 stocks. Results of both experiments are encouraging. Both *PMs* managed to outperform all other strategies by large quantities. *PM2* seems to be a better approach than *PM1*, but we still need to do further investigation.

Table 3: Final Wealth of Portfolio Manager *PM1* and *PM2* with **All 11 Stocks** from a April 1996 to 15 May 2002 (1,539 trading days) Versus *Bank*, *Buy-And-Hold* and *Price-Trend*, Holding at 2% and 5% with 2 Commission schemes. Initial Investment: $10,000.

Final Wealth	Hold @ 2%	Hold @ 5%	Hold @ 2%	Hold @ 5%
	10USDComm	10USDComm	0.1%-Comm	0.1%-Comm
Bank	16,271	16,271	16,271	16,271
B&H	28,617	28,617	30,812	30,812
P-Trend	145	145	4,410	4,410
PM1	207,029	292,676	241,241	360,518
PM2	159,475	292,049	443,690	367,639

Table 4: Final Wealth of Portfolio Manager *PM1* and *PM2* with **9 Stocks** from 26 March 1990 to 15 May 2002 (3,058 trading days) Versus *Bank*, *Buy-And-Hold* and *Price-Trend*, Holding at 2% and 5% with 2 Commission schemes. Initial Investment: $10,000.

Final Wealth	Hold @ 2%	Hold @ 5%	Hold @ 2%	Hold @ 5%
	10USDComm	10USDComm	0.1%-Comm	0.1%-Comm
Bank	26,325	26,325	26,325	26,325
B&H	380,504	380,504	445,400	445,400
P-Trend	3	3	1,688	1,688
PM1	1,352,204	2,526,288	1,746,484	3,340,941
PM2	2,228,379	4,326,860	1,083,689	7,154,038

The total wealth obtained if we were to take the recommendation of the traders shown in Table 2 in equal amounts would be $837,064, profit of 8,271%. Similarly, if we subtract the two shorter series (*nok* and *lu*) and average the resulting wealth over these 9 stocks only, the final wealth would be $963,770, profit of 9,538% – note that the length of these 9 series is slightly greater than the 3,058 trading days used in the *PM* approach.

When using the simple *PM2* approach explained in section 6 with these 9 stocks, we see that the final wealth obtained is $7,154,038, which yields a profit of 71,440% over the original $10,000 investment, much higher than the profit of 9,538%. Therefore these results suggest that apportioning the funds this way yields substantially better results (649% increase!) than using the system with no movement of money between stocks.

Figure 2 shows the wealth during the period examined. The *buy-and-hold* portfolio (initial investment is divided equally between all 9 stocks on day 1 and no more transactions are executed thereafter) of both commission schemes peaked at around

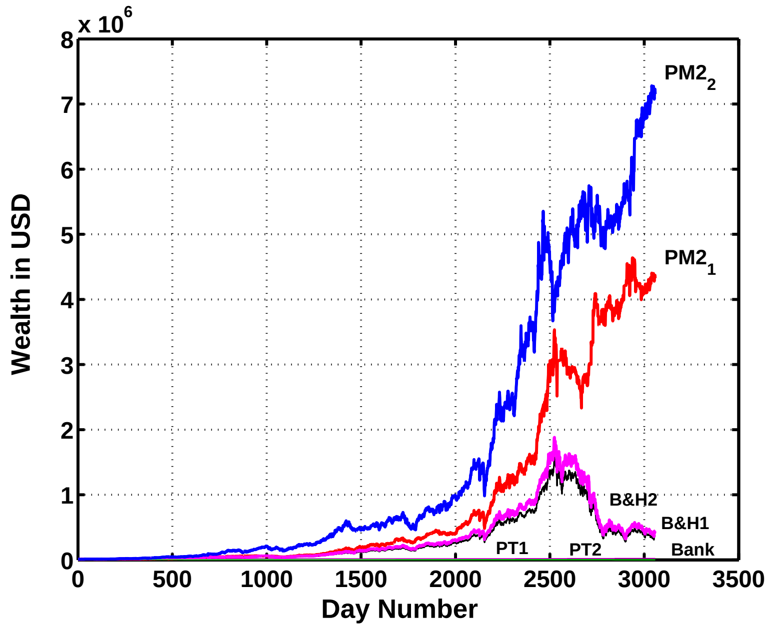

Figure 2: Wealth in USD of *Bank*, *Buy-And-Hold*, *Price-Trend* and *PM2*, Holding at 5% with 2 Commission schemes (from Table 4). Portfolio of **9 Stocks** from 26 March 1990 to 15 May 2002 (3,058 trading days). Initial Investment: $10,000. Final Wealth of *Bank:* $26,326, *B&H1* (Comm. 10 USD): $380,504, *PT1* (Comm. 10 USD): $3, $PM2_1$ (Comm. 10 USD): $4,326,860, *B&H2* (Comm. 0.1%): $445,400, *PT2* (Comm. 0.1%): $1,688 and $PM2_2$ (Comm. 0.1%): $7,154,038.

day 2,523 (end of March 2000); then it started to drop substantially until day 2,781 when the portfolio was valued at $325,846 USD. From that day (beginning of April 2001) until the end (May 15 2002), it has been more or less stable, with several ups and downs balancing out. However, some sectors suffered more than others and the *adaptive agents* of both portfolio approaches, $PM2_1$ and $PM2_2$ managed to get out of the stocks that dropped the most, quickly transferring the money into stocks such as *pep* which was the best performer of all (increasing around 40% in the past two years). However, *PM* increased more than *pep's* 40%, indicating that the traders also learned to exploit certain market irregularities.

7 Conclusions

Portfolio Management is a very challenging task; it is necessary not only to spread the risk of unexpected events such as the attacks on September 11th and Enron's collapse, but it can also be used to outperform other widely used investment strategies such as *buy-and-hold*. The preliminary evidence presented in this paper points to the result that

Portfolio Management demonstrates significantly more forecasting skill than the other methods we have shown. However, as mentioned earlier, we are currently conducting more experiments to validate this outcome.

In this simple model of *artificial traders* we have shown how the best performers were able to survive and furthermore, outperform the *bank*, *buy-and-hold*, *trend-following* and *random* strategies in all the stocks analysed. Clearly the agents were successful in finding profitable rules exploiting the market inertia. For example, making money by selling before the stock price dropped, until trend changes were anticipated, when they started buying again. An adaptive system of this kind seems to be able to at least represent competent traders; and technical indicators, as well as other inputs, proved to be useful in this model due to the heterogeneity of beliefs that exists in real markets.

This economic model is limited in the sense that it captures only a small fraction of the whole repertoire of a real trader's behavior, where the total number of strategies analyzed is, indeed, far greater and more complex than the simple market model used here. However, this system offers perhaps one of the fastest ways of intelligently exploring vast amounts of information, an imperative property which adds great computational complexity when attempting to model modern financial markets. We recall that the idea supported here is that there should be no need to learn market behavior by passing the data more than once. In real life this hardly happens, traders do not experience the same situation again and again, the world is a very dynamic system where one learns to generalize with very limited examples and by trial and error. The argument is that circumstances never repeat exactly and therefore there is no need to repeatedly be faced with previous examples to be tested on future unknown states. We propose that the generalization that one must try to achieve is one of a higher level, in a context that is made throughout the whole life-span of the agent rather than through separate training and testing phases.

8 Possible Applications

Even though it might seem like a very ambitious idea at first, an on-line learning system of this type shows potential to be used as a fully automated trading tool for investment decisions in financial markets. However, this potential use brings the assertion of whether human traders could eventually be replaced by modern systems, which, indeed, is a delicate one and needs to be considered with care. An interesting report entitled *Views from the Frontier: Commentary on the New World of Forecasting and Risk Management* from Olsen and Associates, cites examples where autonomous machine-based systems are already managing large amounts of capital, with the idea that a "number of people believe that human traders, with their limited information processing powers and susceptibility to emotion and unscientific ideas, are too fallible for modern markets." [Ols96]. According to this report, Andrew Lo believes that autonomous trading systems are only feasible in markets that are largely populated by human traders. Lo adds: "you will never be able to replace human interaction until you reach the point where machines become self-aware, and we might never achieve that lofty goal... the aim of the new generation of systems is to augment, rather than replace, human intelligence." This last statement brings the proposed system to another possible use: a decision-making tool.

In general, most AI systems in finance still require a great deal of heuristic knowledge. Such knowledge is provided by financial experts in the form of rules, knowledge bases, data selection and proper manipulation, which strongly suggests that they are best used to complement the decision process of existing team of experts rather than on their own. "Thus they exist more in the realm of statistical tools than 'artificially intelligent' agents. Nevertheless, they are powerful techniques and as our development of them progresses, it is likely that they will find greater and greater utilization on Wall Street" [Gilbert 95]. Such is the case of the system AXON [Ganesh & Barr 94], OMNI [Barr & Mani 94], Advanced Investment Technology's (AIT) patented NN System[4], and many others that have been commercially employed. However, the use of successful, completely automated trading systems is not far from real. A number of successful systems have been developed, such as Prediction Company's[5] fully automated trading system, which builds consistent predictive models of markets and behavior of financial instruments to make decisions based in vast amounts of data; executing transactions based on those decisions in real time.

So far, this model works entirely autonomously, i.e. learns and adapts without any human intervention nor any initial clues given to it, and results shown with no prior knowledge about the market are encouraging, strongly suggesting that it is feasible to model a trader's behavior in such a simplistic way. However, if available, it would be worthwhile to use knowledge from the *real experts* rather than – or in addition to – random initial strategies.

The goal of many decision-making systems is to try to replicate the knowledge (heuristics) of a human in a particular field of expertise. The principle behind these systems is that, in an ideal situation, the human expert's wisdom can be reduced to a series of interconnected generalised rules. However, there are distinct limitations to the abilities of experts in any field to articulate the rules they follow when displaying their expertise. The expert may have no conscious access to much of his expertise. These systems are well suited for problems where there is a consensus of expertise and where there is value of retaining such expertise in a system, such as medical diagnosis. In trading there seems to be no consensus on theories and as a result, many different investment views are applied, even within the same investment group. However, a system of this type could be used to test and develop a variety of strategies given by a single or multiple experts (provided there is availability of data). Here the outcome would be a set of new, improved strategies evolving from the original set, and *real traders's* performance could be tested against this system's. Regardless of a random or an expert start, this system could also be used as a training devise for novice traders.

Other application domains of the present model include a wide variety of areas

[4]Advanced Investment Technology currently manages over $750 million in assets, making use of United States Patent No. 5,761,442, for stock selection. AIT uses nonlinear methods in forecasting, in conjunction to other methods, to confirm stock pricing forecasts and to do back-testing in order to validate their techniques. Web page: http://www.ait-tech.com/

[5]Prediction Company was founded in 1991 by Doyne Farmer, Norman Packard and Jim McGill and works exclusively for UBS Warburg, the investment banking division of UBS AG. Web page: http://www.predict.com/

where there is time series data available, decisions need to be considered frequently and there is some kind of feedback possible about the quality of decisions. For example, there might be applications in the insurance market, in credit scoring, in fraud detection, in marketing prospect assessment and so on. In particular, the foreign exchange market is a very good candidate to test the model. In addition, this approach could be used in various ways: to automate some dealing, to provide a benchmark for use in developing more knowledge-intensive dealing systems, as a training aid for new dealers and as part of a portfolio management system.

However, putting a system of this kind in practice *commercially* requires additional research work and money; i.e., feeding it with *real time data* and allow it to perform intra-daily transactions, licensing the software and customising the agents for every client. So far the idea seems promising, and further research is being carried on in this new and emerging area of *evolving artificial traders in industry*.

References

[Arthur 92] W. Brian Arthur. On Learning and Adaptation in the Economy. Working Paper 92-07-038, Santa Fe Institute, 1992.

[Barr & Mani 94] Dean S. Barr and Ganesh Mani. Using Neural Nets to Manage Investments. *AI EXPERT*, pages 16–22, February 9th 1994.

[Beltrametti *et al.* 97] Luca Beltrametti, Riccardo Fiorentini, Luigi Marengo, and Roberto Tamborini. A Learning-to-Forecast Experiment on the Foreign Exchange Market with a Classifier System. *Journal of Economic Dynamics and Control*, 21(8-9):1543–1575, 1997.

[Chen & Yeh 96a] Shu-Heng Chen and Chia-Hsuan Yeh. Genetic Programming and the Efficient Market Hypothesis. In John R. Koza, David E. Goldberg, David B. Fogel, and Rick L. Riolo, editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 45–53, Cambridge, MA, 1996. The MIT Press.

[Chen & Yeh 96b] Shu-Heng Chen and Chia-Hsuan Yeh. Toward a Computable Approach to the Efficient Market Hypothesis: An Application of Genetic Programming. *Journal of Economic Dynamics and Control*, 21(6):1043–1064, 1996.

[Chen & Yeh 99] Shu-Heng Chen and Chia-Hsuan Yeh. Genetic Programming in the Agent-Based Modeling of Stock Markets. In David A. Belsley and Christopher F. Baum, editors, *Proceedings of the Fifth International Conference on Computing in Economics and Finance*, Boston College, MA, USA, 1999.

[Ganesh & Barr 94] Mani Ganesh and Dean Barr. Stock-specific, non-linear neural net models: The AXON System. In *Proceedings of the Neural Networks in the Capital Markets Conference*, November 1994.

[Gilbert 95] Jeremy Gilbert. Artificial Intelligence on Wall Street. Working Paper, Brandeis University, Department of Computer Science, 1995.

[Lo & MacKinlay 99] Andrew W. Lo and A. Craig MacKinlay. *A Non-Random Walk Down Wall Street*. Princeton University Press, 1999.

[Malkiel 99] Burton G. Malkiel. *A Random Walk Down Wall Street*. W.W. Norton & Company, 6th edition, 1999.

[Ols96] Views from the Frontier: Commentary on the New World of Forecasting and Risk Management – What's Ahead? Report, Olsen & Associates Research Group, May 1996. Interview with Andrew Lo, Blake LeBaron, Andreas Weigend, Martin Peter and other experts.

[Palmer *et al.* 94] R. G. Palmer, W. Brian Arthur, John H. Holland, Blake LeBaron, and P. Tayler. Artificial economic life: a simple model of a stockmarket. *Physica D*, D 75:264–274, 1994.

[Schulenburg & Ross 99] Sonia Schulenburg and Peter Ross. An Evolutionary Approach to Modelling the Behaviours of Financial Traders. In *Genetic and Evolutionary Computation Conference Late Braking Papers*, pages 245–253, Orlando, Florida, 1999.

[Schulenburg & Ross 00] Sonia Schulenburg and Peter Ross. An Adaptive Agent Based Economic Model. In Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson, editors, *Learning Classifier Systems: From Foundations to Applications*, volume 1813 of *Lecture Notes in Artificial Intelligence*, pages 265–284. Springer-Verlag, Berlin, 2000.

[Schulenburg & Ross 01] Sonia Schulenburg and Peter Ross. Strength and Money: An LCS Approach to Increasing Returns. In Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson, editors, *Advances in Learning Classifier Systems*, volume 1996 of *Lecture Notes in Artificial Intelligence*, pages 114–137. Springer-Verlag, Berlin, 2001.

[Schulenburg & Ross 02] Sonia Schulenburg and Peter Ross. Explorations in LCS Models of Stock Trading. In Pier Luca Lanzi, Wolfgang Stolzmann, and Stewart W. Wilson, editors, *Advances in Learning Classifier Systems*, volume 2321 of *Lecture Notes in Artificial Intelligence*, pages 150–179. Springer-Verlag, Berlin, 2002.