The Co-evolution of Trading Strategies in A Multi-agent Based Simulated Stock Market Through the Integration of Individual Learning and Social Learning

Graham Kendall
School of Computer Science and IT
ASAP Research Group
University of Nottingham
Nottingham, NG8 1BB
gxk@cs.nott.ac.uk

Yan Su
School of Computer Science and IT
ASAP Research Group
University of Nottingham
Nottingham, NG8 1BB
yxs@cs.nott.ac.uk

Abstract *– In this paper we present a multi-agent based model of a simulated stock market within which active stock traders are modelled as heterogeneous adaptive artificial agents. We employ the approach of integrating individual learning and social learning to co-evolve these artificial agents with the aim of evolving successful trading strategies. The proposed model was tested on the British Petroleum (BP.L) share from the LSE (London Stock Exchange). Throughout the experiment we see successful trading strategies emerge among the artificial traders. These artificial agents also demonstrate rich dynamic learning behaviours during the simulation. On average, 80% of the artificial stock traders were able to trade using successful trading strategies which brings the investors higher returns compared to a baseline buy-and-hold strategy.*

Keywords *– Multi-agent System, Simulated Stock Market, Trading Strategies, Artificial Neural Network (ANNs), Genetic Algorithm (GA), Individual Learning, Social Learning, Co-evolution.*

1. Introduction

Traditionally the stock market has been studied using standard representative agent models without taking into account the nature of the market where heterogeneous investors with various expectations and different levels of rationality interact with each other through the market. Palmer et al. [1] described a simple multi-agent based model of a stock market inside which independent adaptive agents can buy and sell stock on a central stock market. Based on this idea, various types of Artificial Stock Market (ASM) were developed [2,3,4] and they became more and more important in the study of the stock market – see [5] for a good review on early work on agent based computational financial markets and [6] for the recent advances in evolutionary computation in economics and finance. These multi-agent based ASM models, rather than taking real data from the real world markets, build the artificial stock markets from the ground up using a certain market structure together with the artificial stock traders modelled as heterogeneous adaptive agents. Inside these artificial stock markets, stock prices are generated endogenously and the resulting time series and market dynamics are studied [2,3,4].

Schulenberg et al. [7,8] took another approach by introducing real market data into an adaptive agent based stock market model. They showed that their artificial agents, by displaying different and rich behaviours, are able to discover and refine novel and successful sets of market strategies that outperform a traditional buy-and-hold strategy and risk-less bond. In Schulenberg et al's model, artificial investors are modelled using Learning Classifier Systems (LCSs). One major problem with LCS systems is that the classifier rules are designed explicitly before the evolutionary process of the LCSs begins, thus the novelty of evolved market strategies (LCSs) is questionable.

The other problem, both with Schulenberg et al's model and other early multi-agent based ASM models, is the ambiguity of the difference between individual learning and social learning within these models. Vriend [9] discussed the essential difference between individual and social learning, and its consequences for computational analysis using the experiments carried out in a standard Cournot oligopoly game. Vriend states that "…*the computational modelling choice made between individual and social learning algorithms should be made more carefully, since there may be significant implications for the outcomes generated.*" Chen et al. [4] embraced Vriend's research into their artificial stock market models, and demonstrated that different learning mechanisms resulted in little difference in the macro-structures, i.e. the econometric properties of the time series of the generated artificial stock markets. However, different learning mechanisms generated different micro-structures of the resulting artificial stock markets regarding the traders' behaviour and belief.

Our aim here is to employ Chen et al's approach, and apply it to the real world stock market. We propose a multi-agent based simulated stock market where market scenario, such as stock price and trading volume, are given exogenously. Inside the simulated stock market, heterogeneous artificial stock traders, modelled using artificial neural networks, will trade stocks using real market data and co-evolve with each other by the means of individual and social learning. Our current experiment, testing our model on the British Petroleum (BP.L) share from the London Stock Market, shows that, 80% of the artificial stock traders outperformed the baseline buy-and-hold strategy and the artificial agents demonstrate rich dynamic learning behaviours.

2. Background

Chen et al. [4] discussed the two main differences among the agent-based approaches for studying financial markets: representation of agents and learning mechanism. In Schulenberg et al.'s experiments [7,8], three different types of traders with pre-defined types were studied. We intend to break the constraints on these pre-defined traders by representing our artificial traders using randomly generated artificial neural networks (ANNs). Traditionally, artificial stock traders modelled using ANNs tend to use the same set of indicators from the market which is contradictory to the fact that different people in the market receive different sets of information from the market. To solve this problem, we propose a central pool of technical indicators from which traders will select indicators to form different types of trading strategies.

This central pool is also the mechanism through which the social learning process is carried out. This central pool, in fact, is a simulation of the social culture in the simulated market. Traders are allowed to tell other traders how important he believes his indicators are by assigning scores them. Traders are also allowed to publish their successful strategies into the central pool so that other traders can learn his strategy.

3 The Model

3.1 Simulated Stock Market

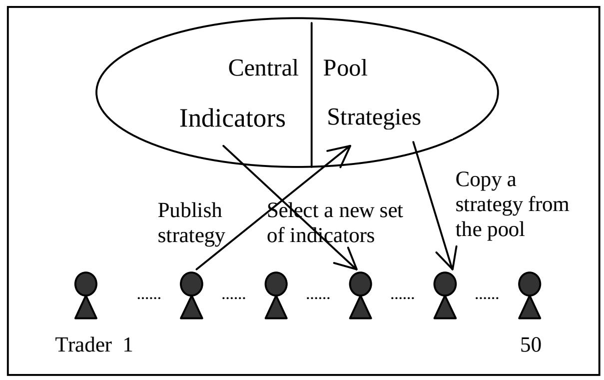

Fig. 1. Simulated Stock Market

Figure 1 shows our multi-agent based model of a simulated stock market, which is described as follows:

1. Before trading starts, there are 50 active traders in the simulated stock market. There are 20 indicators and zero trading strategies in the central pool. The 20 available indicators are assigned an equal score of 1. Each trader selects a random number of indicators using roulette wheel selection.
2. With the set of indicators selected, each trader generates ten different models. These ten models may have different network architectures, but they use the same set of indicators selected by the trader. The aim is for the trader to evolve models from these ten by the means of individual learning.
3. The time span of the experiment covers 3750 trading days, which is divided into

30 intervals. Each interval contains 125 days (6-month trading).

4. Each 125-day trading is sub-divided into intervals of 5 days. Each trader trades for 5 days, and then undertakes individual learning by means of a Genetic Algorithm (GA).
5. At the end of each 125-day trading, social learning occurs and each trader is given the opportunity to decide whether to look for more successful strategies from the pool or whether to publish his/her successful strategies into the central pool.
6. After social learning has finished, the system enters the next 125 trading days and steps 4, 5 and 6 are repeated.
7. For every transaction, buy means use all the cash in the trader's account and sell means sell all his holdings. Both margin account, where traders could buy stocks on credit, and short selling, where traders could sell stocks she/he does not hold, and buy it back at a later time, are not allowed. Traders are asked to pay a trading fee of £10 for each transaction. Traders are also paid interest for any cash in their account, with an annual interest rate of 5%. Interest is calculated every half year.

Except the 50 active stock traders, there is also one investor using a traditional buy-and-hold strategy and one investor who saves all the money in a bank. Their performance will serve as benchmarks for the 50 active traders. The buy and hold investor will use all the money in the bank to buy the stock on the first trading day, and hold it until the last day of trading. The bank savings investor will sell all shares on hand on the first trading day, and keep all the money in a bank for the entire period, receiving an annual interest rate of 5%. On the first trading day, all traders and investors are given a portfolio of £100,000 cash in bank and 1000 BP shares.

3.2 Data and Data Pre-processing

Shares of BP PLC from the London Stock Market is selected to be traded in the simulated stock market. Fig 2 shows BP's historical price.

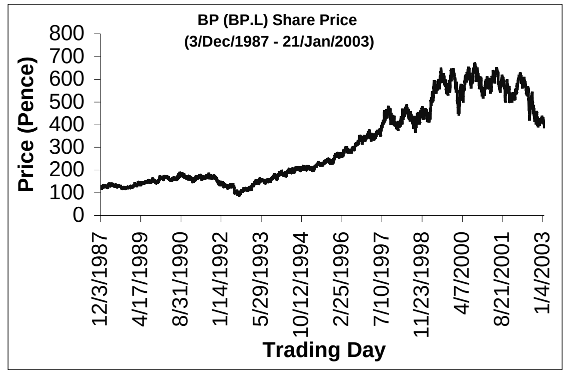

Fig. 2. BP PLC (BP.L) share price

Besides the primitive historical share price, other financial data is also used to compose 20 popular technical indicators. This data includes: trading volume; intra-day high, intra-day low; FTSE-100 index; DJ Oil&Gas Index(UK), S&P 500 Index and DJ INDU AVERAGE. All data was acquired from Yahoo Financial (http://uk.finance.yahoo.com/). Table 1 shows the 20 technical indicators used.

Table 1. Technical indcators that are used as inputs into the neural networks. All values are normalised into the range of [0,1].

TI	Description
1	10 days moving average
2	20 days moving average
3	50 days moving average
4	200 days moving average
5	Closing price (normalized)
6	Rate of change (price)
7	Oscillator (price)
8	10 days bias
9	20 days volume rate of change
10	10 days relative strength
11	14 days relative strength
12	21 days relative strength
13	Stochastic oscillators (k%)
14	Fast stochastics (D%)
15	Slow stochastics (slow D)
16	FTSE-100 Index rate of change
17	Relative strength index to FTSE-100 Index
18	S&P 500 Index rate of change
19	DJ INDU AVERAGE index rate of change
20	DJ Oil&Gas Index (UK) rate of change

4. GA and Individual Learning

4.1 Prediction Model

The neural networks used by the traders are multi-layer feed-forward networks. The networks are either 2-layer (no hidden layer) or 3-layer (one hidden layer). Two different types of activation function (sigmoid and tanh) are used. There is one single output node from the network. In order to facilitate the GA learning process, the description file of each neural network is designed in a way such that it can also be used as a chromosome within the GA, as shown in Fig 3.

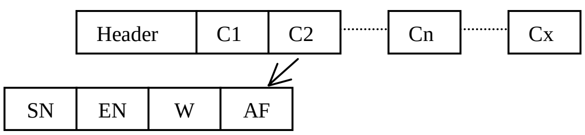

Fig. 3. A neural network chromosome. Each chromosome consists of a header and a number of connections. The header contains general information about the network: starting input node, ending input node, starting hidden node, ending hidden node. Each connection, Cn, contains four components: starting node (SN), ending node (EN), weight (W), and activation function (AF). During the GA process, both the weights of the connection (W) and activation function (AF) are mutated.

Besides the mutation of weights and activation function, the structure of network is also evolved by means of adding a new node or deleting a node from the chromosome. SN and EN are used to keep track of the order of connections in the neural network.

As stated above, traders are allowed to use different sets of indicators for trading. Table 2 shows the number of indicators used by trader no. 1 to traders no. 24 on the first day of trading.

Table 2. Number of indicators (NOI) used by trader no. 1 to no.24 on the first trading day.

Trader	NOI	Trader	NOI	Trader	NOI
1	18	9	15	17	2
2	3	10	16	18	18
3	8	11	14	19	17
4	2	12	18	20	2
5	14	13	14	21	16
6	3	14	10	22	5
7	8	15	1	23	14
8	6	16	12	24	2

4.2 Individual learning

Individual learning occurs during every 125-day trading period. At the start of each period, each trader decides which set of indicators they will use to build their prediction models. Each trader builds ten models based on their selected indicators. These ten models all use the same set of indicators, but with different network architectures. Each trader evolves his ten models in an attempt to achieve better prediction models, using a GA described below.

During the 125 trading days, a model is chosen, using roulette wheel selection, for the next 5 days trading. The selection is based on the ten models' scores. At the end of each 5-day trading, trader's ROP (rate of profit) is calculated using Formula 1.

$$ROP = \frac{W - W^{'}}{W^{'}} \times 10 \qquad (1)$$

W is the trader's current assets (cash + shares). $W^{'}$ is the trader's assets one week before. The selected model's score is then update using Formula 2.

$$m_i^n = m_i^n + ROP \qquad (2)$$

where i is trader i and n is the n^{th} model selected from the 10 models. Based on the new updated scores, four models are selected as parents, using roulette wheel selection. Another four models, those with the lowest scores, are selected and will be replaced by four new offspring (produced by the four parents through mutation). Overall, the four parent models selected and the two remaining models will stay intact and continue to the next generation together with the four new offspring.

As a trader's prediction models (neural networks) has different numbers of hidden nodes, possibly different numbers of hidden layers and maybe uses different activation functions, it will not be sensible to use a crossover operator in the GA. Therefore, within the GA we set the probability of crossover 0 and mutation to 1. The complete individual algorithm is given in Figure 4:

```
Select models to be mutated using
   roulette selection;
Select models to be eliminated;
Decide number of connections to be
   mutated, m;
i = 0;
While(i < m){
      Randomly select a connection;
```

```
      Weight = weight + Δw;
      i = i + 1;}
   With 1/3 probability add hidden node;
   With 1/3 probability delete hidden
     node;
   replace models to be eliminated with
     the new mutated models;
```

Fig. 4. Individual learning

The number of connections to be mutated, *m*, is a random integer between 0 and the total number of connections in the selected neural network. Δw is a random Gaussian number with a mean of zero and standard deviation of one. Besides the mutation of weights, we also evolve the structure of the network by allowing the probability of adding or deletion of hidden nodes. After producing ten new models, the trader will select a model for the next 5 trading days, using roulette wheel selection. Individual learning occurs at the end of every 5-day trading for each trader.

5. Social Learning

After 25 weeks (125 days) of trading and individual learning, all traders enter a social learning stage. During social learning, all traders have the chance to see how other traders are performing. Traders may decide to learn from other traders, or publish their own successful trading strategies. At this stage, each trader will carry out a self-assessment. The trader's decision in social learning depends on the result from this self-assessment. Based on the methods used by Chen et al. [4], our trader's assessment is calculated using Formula 3, 4 and 5. First, the traders' rate of profit (ROP) (Formula 1) for the past six months is calculated, and the 50 traders are ranked from 0 to 49 according to their ROP.

$$S^i_{peer} = 1 - \frac{R_i}{49} \qquad (3)$$

R_i is the rank of trader i in the range of [0,49] (0 means highest rank with largest ROP). Formula 3 gives each trader a score in terms of peer pressure from other traders. In other words, this score shows trader *i*'s performance compared to other traders.

$$S^i_{self} = \frac{ROP - ROP'}{100} \qquad (4)$$

ROP is the rate of profit for the current six months trading. ROP' is the rate of profit for the previous six months. Formula 4 gives the trader's score in terms of his own performance in the past six months compared to the previous six months. Finally, these two types of performance are composed into Formula 5, which gives the overall assessment for trader *i*.

$$assessment_i = S^i_{peer} + \frac{1}{1+e^{(1-S^i_{self})}} \qquad (5)$$

The final assessments for 50 traders are then normalised into the range of [0,1]. Depending on their assessment, a trader may choose to:

1) If a trader's assessment is 1, and the trader is not using a strategy drawn from the pool, then publish the strategy into the central pool. Go into the next six months trading using the same strategy.
2) If a trader's assessment is 1, and the trader is using a strategy copied from the pool, do not publish it again, but update this strategy's score in the pool using their six month ROP. Go into next six months trading using the same strategy.
3) If a trader's assessment is less than 0.9, the trader has 0.5 probability of copying a strategy from pool, which means the trader will discard whatever model he is using, and select a better trading strategy from the pool using roulette selection, and go into the next six months trading with this copied strategy. Or, with 0.5 probability, the trader will decide to discard whatever strategy he is using, and select another set of indicators as inputs, build 10 new models and go into next six months trading with these 10 new models.
4) If assessment is between 1 and 0.9, the trader is satisfied with his performance in past six months and continues using that strategy.

Traders will also update scores of indicators they have used in the central pool based on their performance in the current six months using Formula 6 below.

$$I^n_i = I^n_i + ROP \qquad (6)$$

where *i* is the trader i. *n* is the n^th indicator used by trader i in the current six month trading. *ROP* is the rate of profit of the trader i in the current six months trading.

6. Experimental Result

The main consideration of choosing the BP share as the test bed for our model is that his share is a upturn stock in the overall trend, see figure 2. A buy-and-hold strategy will, obviously, bring the investor a positive return. If our artificial traders could achieve a higher return than the classical buy-and-hold strategy through both their individual and social learning, that means these artificial agents have discovered successful trading strategies during the evolution. The results from our experiment proved successful trading strategies have been developed.

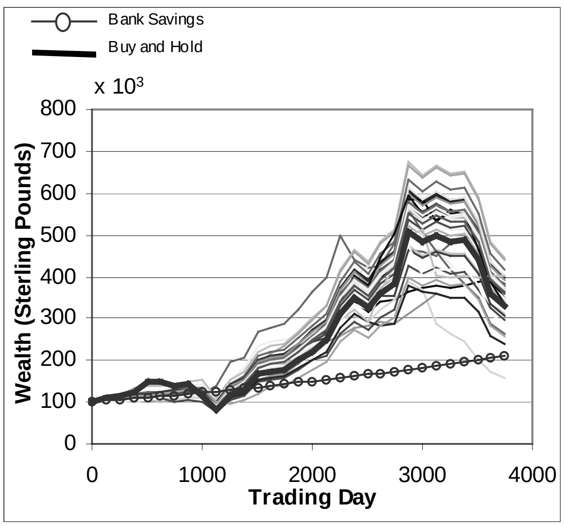

Fig. 5. Traders' Performance (BP.L share)

Figure 5 demonstrates the growth of wealth of our 50 artificial traders during the 3750 days trading period. The thick black line indicates the growth of wealth of the investor with buy-and-hold strategy. The thin line at the bottom of the diagram with o markers indicates the growth of wealth of the investor who saved his money in bank for 15 years. From Figure 5 we can see the majority of the artificial traders (represented by the lines above the thick black line) were able to learn to predict the trend in the stock, i.e. start to buy in stocks when the share is going up and start to sell it when it is going down. The more accurately the trader was able to control the timing of buying and selling, the faster they accumulated wealth. Table 3 gives the statistic results on the 50 artificial traders. It shows 40 out of the 50 artificial traders beat the buy and hold strategy. On average, 50 active traders outperform buy and hold strategy by 25.84%.

Table 3. Results from 50 traders compared with buy and hold strategy and bank savings. All returns are calculated as the difference between wealth on day 3750 and the investor's original wealth divided by the original wealth.

Description	Result
Return from bank savings	109.76%
Return from buy-and-hold strategy	228.69%
Average return from the 50 traders	254.53%
Maximum return among the 50 traders	338.53%
Minimum return among the 50 traders	57.7%
No. of traders who outperform savings	49
No. of traders who outperform buy and hold	40

To see the rich learning dynamics of 50 artificial agents more clearly, we selected 3 traders from the 50 traders and depict them in Figure 6. Trader 14 is the best performer who achieved a return of 338.53% on his original wealth. Overall, trader 14 was able to predict the trend in the price of the stock fairly well. The transaction records of trader 14 shows this trader, in fact, learned a strategy from the central pool in the early days, and kept it until the last trading day. Some other traders also used the same strategy copied from the pool, but trader 14 refined this strategy constantly through his own individual learning which results in his outstanding performance compared to others.

Trader 16 is the worst performer whose wealth line finally runs below bank saving line. This trader, showed by his transaction records, did not consult the central pool for other trader's strategy throughout the whole trading period. He basically followed the buy-and-hold strategy in the early stage of the trading period. The trader's own individual learning did not help him too much. Around day 2200, he made a mistake by selling the stock when the price was still going up. When the stock price dropped dramatically around day 3000, this trader's strategy completely failed. On the contrary, trader 2, shown by his transaction records, constantly searched for good strategies from the pool, and tried out different strategies during the different stage of the trading period. Before day 2300, trader 2 used a strategy learned from central pool which worked quite well during the upturn period. However, during the downturn after day 2300, this strategy didn't work very well. Trader 2 went on and tried different strategies from the pool, finally still managed to outperform buy-and-hold strategy.

In summary, our artificial agents demonstrated rich dynamic and interesting learning behaviours during the simulation which is very similar to real stock traders in the real world market. The mechanism of integrating individual and social learning here played an important role in the sense of agents' learning behaviours and abilities.

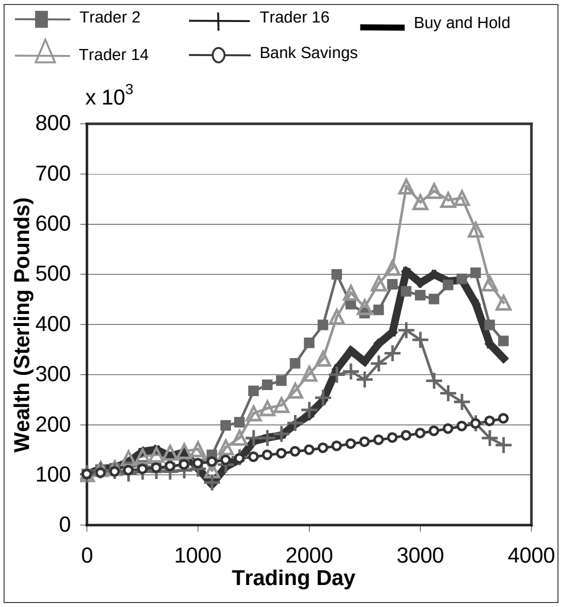

Fig. 6. Traders' Learning Dynamics

7. Conclusion

Compared to the traders with specifically pre-defined types studied in Schulenberg et al. [7,8], our 50 artificial traders were generated completely randomly without defining what types of rationality or belief they should have. These 50 artificial stock traders, imitating real world traders, traded a stock in a simulated stock market, learned to trade by themselves and learned from other traders through social learning. The results from the simulation shows 80% (40/50) of our artificial agents learned successfully in trading stock, and outperformed the baseline buy-and-hold strategy. Moreover, these 50 randomly generated artificial traders demonstrated more dynamic and interesting learning behaviours during the simulation compared with the three different types of traders studied in Schulenberg et al.'s experiments. It will be very interesting to see how different learning mechanisms, for example, a solo individual learning process, a solo social learning process, compared to a integrated individual and social learning process, affect the artificial agents' learning behaviours and abilities. This is one the direction of our future research.

References

[1] Palmer, R.G., Arthur, W.B., Holland, J.H., LeBaron, B., Tayler, P.: Artificial economic life: a simple model of a stock market. Physica D, Vol. 75. (1994) 264-274

[2] Arthur, W.B., Holland, J.H., Lebaron, B., Palmer, R., Tayler, P.: Asset Pricing under Endogenous Expectations in an Artificial Stock Market. The Economy as an Evolving Complex System, XXVII. Addison-Wesley (1997) 15-44

[3] Lebaron, B., Arthur, W.B., Palmer, R.: Time Series Properties of an Artificial Stock Market Model. Journal of Economic Dynamics and Control (1999), 1487-1516

[4] Chen, S.H., Yeh, C.H.: Toward an integration of social learning and individual learning in agent-based computational stock markets: The approach based on population genetic programming. Journal of Management and Economics, 5, (2001)

[5] Lebaron, B.: Agent based computational finance: Suggested readings and early research. Working paper, Graduate School of International Economics and Finance, Brandeis University, October (1998)

[6] Chen, S.H.: Evolutionary Computation in Economics and Finance, Physica-Verlag, (2002)

[7] Schulenberg, S., Ross, P.: An Adaptive Agent Based Economic Model. In: Lanzi, P.L., Stolzmann, W., Wilson, S.W. (eds): Learning Classifier Systems: From Foundations to Applications, Vol. 1813 of LNAI. Springer-Verlag, (2000) 265-284

[8] Schulenburg, S., Ross, P.: Strength and Money: An LCS Approach to Increasing Returns. In Lanzi, P.L., Stolzmann, W., Wilson, S.W., (editors). Advances in Learning Classifier Systems, volume 1996 of Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin, (2001), 114 - 137

[9] Vriend, N.: An illustration of the essential difference between individual and social learning, and its consequences for computational analysis. Technical report. Queen Mary and Westfield College, University of London (1998)