An Artificial Stock Market

R.G. Palmer, W. Brian Arthur, John H. Holland, and Blake LeBaron
Santa Fe Institute
1399 Hyde Park Road
Santa Fe, NM 87501, USA

Abstract

The Santa Fe Artificial Stock Market consists of a central computational market and a number of artificially-intelligent agents. The agents choose between investing in a stock and leaving their money in the bank, which pays a fixed interest rate. The stock pays a stochastic dividend and has a price which fluctuates according to agent demand. The agents make their investment decisions by attempting to forecast the future return on the stock, using genetic algorithms to generate, test, and evolve predictive rules.

The artificial market shows two distinct regimes of behavior, depending on parameter settings and initial conditions. One regime corresponds to the theoretically-predicted rational expectations behavior, with low overall trading volume, uncorrelated price series, and no possibility of technical trading. The other regime is more complex and corresponds to realistic market behavior, with high trading volume, high intermittent volatility (including GARCH behavior), bubbles and crashes, and the presence of technical trading. One parameter that can be used to control the regime is the exploration rate, which governs how rapidly the agents explore new hypothesis with their genetic algorithms. At low exploration rate the market settles into the rational expectations equilibrium. At high exploration rate it falls into the more realistic complex regime. The transition is fairly sharp, but close to the boundary the outcome depends on the agents' initial "beliefs"—if they believe in rational expectations it occurs, and is a local attractor; otherwise the market evolves into the complex regime.

1 Introduction

Today's standard economic theory, general equilibrium theory, or *rational expectations*, says in its shortest statement that agents—traders, firms, individuals, etc—*deduce* their optimum behavior by logical processes from their circumstances. It further assumes that the agents have complete information, that they're perfectly rational, that they have common expectations, and that they know that everyone else has these same properties too.

One of the consequences of this approach is that almost everything is decided at time zero. The agents first work out how the whole future should be, and then the world just plays itself out. There is no dynamics, no learning, and no evolution.

When this rational expectations approach is applied to a stock market [5], it implies that there shouldn't be anything like market moods or psychologies. There shouldn't be bubbles, crashes, or bursts, and volatility should be low. There shouldn't be much trading volume; the only reason one person would trade with another would be if something happened externally, changing the assets available for investment. There shouldn't be money to be made by *technical trading*, simply extrapolating patterns in a time series of price, because any regularity in the price series should have already been arbitraged away by the rational agents.

These ideas don't fit the facts of real stock markets very well. There do seem to be bubbles, crashes, and moods of the market. The volume and volatility are much higher than can be accounted for by external changes. And people on Wall Street do seem to make money by technical trading (see e.g., [4]).

There are some ways to modify the theory or its application to attempt to come to terms with these discrepancies, including a number of ideas of *bounded rationality*, and theories involving *noise traders* [7]. However none of these theories seem wholly convincing [2], and we describe here a very different approach.

In starting this project in 1989 [1], we asked what agents *really* do in markets, and more generally in the world. Our answer then, and now, is that they (a) classify whatever they see; (b) notice patterns; (c) generalize and form internal models, ideas, or rules of thumb; and (d) act on the basis of those internal models. We decided to build a stock market along these lines in the computer, whereby agents would notice the patterns

in the price (and in any other data they had access to), form models, and then trade on that basis.

Of course, the agents have to evaluate and adapt their internal models after seeing how well they work. Actually, each agent has a number of different ways of predicting the future, and is continually evaluating and comparing them. The ones that work well gain more weight and are used more often. The ones that fail are eventually thrown out and replaced.

The agents buy and sell stock in the market, and thereby affect the stock price. What the agents do affects the market. What the market does affects the agents. So the market behavior emerges from the collective behavior of the agents, who are themselves coevolving.

From an economics viewpoint, the aim of our work is to examine a market of interacting agents that can learn with an open set of possibilities, and see whether it converges to a rational expectations equilibrium or to something else. The core result is that there are two equilibria. The model can show rational expectations behavior, and it can show realistic market behavior, but they are two separate equilibria.

2 Structure of the Market

The basic structure of the model is N agents ($i = 1, 2, \ldots, N$) interacting with a central market. Typically $N = 50$–100. There may be several types of agents. In contrast to many other interacting agent models, the agents don't interact directly with each other, but only via the market.

In the market there is a single stock, with price $p(t)$ per share at time t. Time is discrete ($t = 1, 2, \ldots$); period t lasts from time t until $t+1$. The stock pays a dividend $d(t+1)$ per share at the end of period t. The dividend times series $d(t)$ is itself a stochastic process defined independently of the market and the agents' actions. We normally use a simple random process with persistence called an AR-1 or Ornstein-Uhlenbeck process, given by

$$\hat{d}(t+1) = \rho\hat{d}(t) + \alpha\eta(t)$$

where $\hat{d}$ means the offset of the dividend from a fixed mean $\bar{d}$, so $d(t) = \bar{d} + \hat{d}(t)$, ρ and α are parameters, and $\eta(t)$ is a Gaussian random variable, chosen independently at each time t from a normal distribution with mean 0 and variance 1.

There is also a fixed-rate asset, *the bank*, which simply pays a constant rate of return r per period. The agents have to decide how much money they want to put into the stock (which has a fixed total number of shares—if somebody buys, someone else has to sell), and how much they want to leave in the bank. At any time t, each agent i holds some number of shares of stock $h_i(t)$ and has some amount of cash $M_i(t)$ in the bank. Its total wealth is then

$$w_i(t) = M_i(t) + h_i(t)p(t).$$

At the end of the period, one time step later, this portfolio becomes worth

$$\bar{w}_i(t+1) = (1+r)M_i(t) + h_i(t)p(t+1) + h_i(t)d(t+1),$$

where the three terms are the money in the bank, with interest, the new value of the stock, and the dividend pay-out. $\bar{w}(t+1)$ and $w(t+1)$ are not the same, because trading occurs in between, potentially moving assets between the bank and the stock.

The trading process is managed by a *specialist* inside the market. The specialist also has the job of setting $p(t+1)$. Its fundamental problem at each time step is that the number of bids to buy and offers to sell may not match, and yet the total number of shares of stock is fixed. We have explored several approaches to this issue, including rationing of bids or offers [6], having the specialist maintain a buffering inventory itself, and holding an auction in which the price at a given time is adjusted until the bids and the offers match closely. Only the last approach, an auction, is described further here. If there are more bids than offers, then the price is raised, so the bids drop and the offers increase, until they match closely.

One more thing that is defined at the level of the overall market structure is the information that is made available to the agents for use in their decision making. In principle this information set (which we call *the world*) consists of the price, dividend, total number of bids, and total number of offers, at each past time step. There are other variables that we have tried adding too, including a predictor of the future dividend (which can be done in the computer, by running the stochastic process forward, but not in the real world!), and a random "sunspot" variable around which the agents might coordinate their actions.

However, we usually condense most of this information into a string of *world bits*. At any given time, the world that the agents see consists of a string of 80 or so bits, and some recent price and dividend information. Some examples of these bits, each of which is either *true* or *false* at each time t, are as follows:

1. $rp(t)/d(t) > \frac{3}{4}$
2. $rp(t)/d(t) > 1$

3. $rp(t)/d(t) > \frac{5}{4}$
4. $p(t) > \mathrm{MA}_{10}\{p(t)\}$
5. $p(t) > \mathrm{MA}_{100}\{p(t)\}$
6. $p(t) > \mathrm{MA}_{500}\{p(t)\}$
7. Always *true*

Here $\mathrm{MA}_n\{p(t)\}$ means a moving average over the most recent n steps of $p(t)$, i.e.,

$$\mathrm{MA}_n\{p(t)\} = \frac{1}{n}\sum_{k=0}^{n-1} p(t-k)$$

The quantity $rp(t)/d(t)$ would be 1 in a simple equilibrium notion of fundamental value, so the deviation from 1 gives a sense of how much the stock is underpriced or overpriced.

We classify bits into 3 categories: technical, fundamental, and control. Technical bits, by definition, just depend on the past price series, and are the only ones that a strict technical trader would use. Bits 4–6 in the above list are technical bits. Control bits are useless ones that we include as experimental controls, like bit 7 in the above list. Fundamental bits are anything else, generally involving the dividend series in some way. Bits 1–3 above are fundamental.

3 Structure of the Agents

Fundamentally the agents have to decide whether to invest in the stock or the bank. If at any time step, they conclude that they want to invest more in the stock than previously, then they submit a bid to buy more shares. Conversely, they may submit an offer to sell shares.

We have examined many types of agents, and our software can mix different types in the same market (see [6] for a description of *condition-action agents*). But this paper only treats *forecasting agents* that use number of *predictors*, each of which attempts to predict the future return (price plus dividend). By seeing how well their predictors work, the agents can estimate their accuracy (prediction variance) and update or replace poor ones. Because they know the variance of their overall predictions, the agents can also perform a risk aversion analysis called CARA, constant absolute risk aversion. This is a standard computation, based on an exponential utility function and used in portfolio analysis, that gives an optimal division of funds between two possible assets when the mean and variance of the expected return is known for each asset. If agent i's estimate of the mean return is $E_i[p(t+1)+d(t+1)]$ with variance σ_i^2, then under CARA (and an additional Gaussian assumption) the ideal number of shares to hold is given by

$$h_i^{\mathrm{desired}}(t) = \frac{E_i[p(t+1)+d(t+1)] - p(t)(1+r)}{\lambda\sigma_i^2}$$

where λ is a parameter, the degree of relative risk aversion.

The agents' predictors actually consist of two parts, a *condition part* and a *forecast part*. The condition part determines when each particular predictor is *activated*, as explained below. Only activated predictors produce forecasts, using their forecast part. The forecast part, in the simplest case, is just a linear rule

$$E_{ij}[p(t+1)+d(t+1)] = a_{ij}\big(p(t)+d(t)\big) + b_{ij}$$

where $E_{ij}[]$ means the expected (predicted) value for agent i's j'th predictor, and a_{ij} and b_{ij} are the coefficients that constitute the forecast part of this predictor. Although this is itself a very simple linear form, the condition parts make the overall prediction only piecewise linear.

Every time a predictor is activated, the agent checks to see how well it performed when the period is over. This is used to maintain a variance σ_{ij}^2 for each predictor, as a weighted moving average of its past squared errors.

There are several ways to combine a set of predictions and variances, $E_{ij}[p(t+1)+d(t+1)]$ and σ_{ij}^2, for each activated predictor j, into the single overall forecast and variance, $E_i[p(t+1)+d(t+1)$ and σ_i^2, needed for the CARA calculation. The simplest, used for all the results described here, is to use the currently best predictor, the one with the smallest σ_{ij}^2 across all activated j's.

The condition part of each predictor is implemented with a *classifier system*, in which the condition part is represented by a ternary string of the symbols $\{0, 1, \#\}$, one for each of the world bits that the agent can observe (we can restrict agents to see only a subset of all the world bits). A condition symbol of `0` means that the corresponding world bit must be *false* for the condition part to match, while conversely `1` requires *true*. A condition symbol of `#` is a *don't care*, and matches either *true* or *false*. For example, the condition string `##1###0#` matches the world bits `01110100` (where `0` stands for *false* and `1` for *true*), but not `10100110`.

Some of an agent's predictors may give good predictions when they are activated, while others may not.

A genetic algorithm is used to adjust and evolve a better set of predictors. For each agent at each period we run the genetic algorithm with probability $1/K$, where K is a parameter. The genetic algorithm eliminates some of the worst predictors, those that have the highest variance, and generates some new ones to replace them. Typically we replace 20 out of 100 predictors.

To generate new predictors we first clone some of the best existing ones in the current population. Then we either perform *mutation* or *crossover* (or sometimes both) on those cloned predictors. Mutation means changing a few condition bits, and modifying the a_{ij}'s and b_{ij}'s by a random amount. We use parameterized distributions of such changes; see [2] for details. Crossover means selecting two parent predictors, taking some condition bits from each, and interpolating their a_{ij}'s and b_{ij}'s. It is not clear whether crossover has any positive effect beyond causing large jumps in the space of condition bits. We also sometimes perform *generalization*, changing some of the fixed bits (`0`'s and `1`'s) to don't cares (`#`'s) in predictors that have not been activated for a long time.

In choosing predictors for replacement and cloning, we mainly select according to variance; low variance means high fitness. But we also impose a small penalty for each bit that is not a `#`, giving a little pressure not to condition the predictors on too many bits.

4 Results

Our main initial goal in this project was to look for realistic market behavior. We asked in particular whether the price series of our stock would look like a real stock price series. We realized that goal several years ago now, finding reasonably realistic market behavior as shown by time series analysis.

In 1995, we started a second phase of experiments by asking whether our agents could also show rational expectations behavior. If homogeneity is assumed, so that all the agents have the same beliefs, then the rational expectations equilibrium for this market can be computed, and has a simple form in which the price is linear in the dividend. Thus rational expectations behavior is within the framework of our agents' linear forecasts.

We tried giving the agents initial beliefs in the rational expectations result by setting the initial conditions for a_{ij} and b_{ij} to the calculated rational expectations values. We found that they stayed there, and that the rational expectations equilibrium is in fact a local attractor—when we started the agents initially fairly close to it, they went to that state. It resulted in a very stable market, just as the theory says, with very little trading going on, and homogeneous agent behavior.

On the other hand, when we started the agents with almost any other conditions, they never settled into rational expectations behavior even after millions of periods, and the system behaved much more like a real market. Thus we had found two regimes of behavior, which we call the "rational expectations regime" and the "complex regime" respectively.

In the rational expectations regime, we see relatively low trading volume, with very little information in the price series that can be exploited for prediction. The forecast parameters—the a_{ij}'s and b_{ij}'s—all converge to be the same; the agents end up becoming homogeneous. The technical bits are not useful, and are dropped from use.

In the complex regime, we find that the agents remain heterogeneous and continuously co-evolve. One of the tests we did was to take a successful agent out of the market, freeze it, and then reinsert it thousands of periods later. We found that it didn't do well at all. Even though the market looks statistically much the same, an agent that was trained in one period doesn't work well later, because the detailed information that it is picking up in its bits and forecasts is changing over time.

The trading volume remains much higher in the complex regime. It varies greatly, and has GARCH behavior. It is auto-correlated, and there are correlations between volume and volatility. These are all features found in real markets. There are sometimes bubbles and crashes, fairly minor ones usually, and over-reactions. And the agents *do* use the technical bits, despite the small cost to do so.

More recently, in a third phase of experiments reported in [2], we looked at what happens as we change certain parameters. We found that we can force the market into either regime with appropriate parameter values, using the same random initial conditions. But with intermediate parameter values, the initial conditions dictate which regime is reached, as in the second phase of experiments.

In most experiments, we varied just the parameter K that dictates how often the genetic algorithm is run. That controls how often the agents explore new possible ways of predicting the future. We frequently compared two values, $K = 250$ and $K = 1000$, that we call *fast exploration* and *slow exploration* respectively. Fast exploration puts us in the complex regime, while slow exploration gives the rational expectations

regime. We always used random initial conditions in these experiments.

An example of our time series analysis was a fit of the price series to a linear recurrence relation

$$p(t+1) = A + Bp(t) + \varepsilon(t+1)$$

where $\varepsilon(t)$ represents the residual variation after fitting the best values for A and B. In a rational expectations equilibrium the residuals $\varepsilon(t)$ ought to be independent identically-distributed Gaussian variables, because the price is driven by the AR-1 dividend series. So we tested the $\varepsilon(t)$ series for normality and correlations. The following table shows some results for fast and slow exploration, from averages over 25 runs of each case.

	Fast Exploration	*Slow Exploration*
σ	2.147 ± 0.017	2.135 ± 0.008
κ	0.320 ± 0.020	0.072 ± 0.012
ρ_1	0.007 ± 0.004	0.036 ± 0.002
$\rho_i^{(2)}$	0.064 ± 0.004	0.017 ± 0.002

The first line, σ, shows the standard deviation of $\varepsilon(t)$. The second line, κ, shows the excess kurtosis, a measure of non-Gaussian behavior based on the fourth moment. Fast exploration gives much larger deviations from normality, in the direction of the "fat tails" seen in real market data. The third and fourth lines show two measures of single-step autocorrelation, $\rho_1 = \langle \varepsilon(t)\varepsilon(t+1) \rangle$, and $\rho_1^{(2)} = \langle \varepsilon(t)^2 \varepsilon(t+1)^2 \rangle - \sigma^4$.

The ARCH(1) test [3], gives about 37 for the fast exploration case, compared to 3.2 for the slow exploration case.

This sort of analysis can be extended by including additional terms in the recurrence relation. For example to test whether the $rp(t)/d(t) > \frac{3}{4}$ technical bit is actually of any use in predicting the price series, we write

$$p(t+1) = A + Bp(t) + CI_{rp/d>\frac{3}{4}}(t) + \varepsilon(t+1)$$

where $I_{rp/d>\frac{3}{4}}(t)$ is 0 or 1 as appropriate at each t. We then fit the coefficients as before. In this case we found $C = -0.44 \pm 0.10$ for the fast exploration case, compared to $C = 0.05 \pm 0.09$ for the slow exploration case. Thus this bit is useful in the fast but not the slow case.

There is much more yet to explore with this model. Given a whole computational market in the computer, we can experiment with what happens upon changing the market structure, the specialist, the dividend series, and so on. Some other future plans include:

1. Multiple stocks. It is not clear whether introducing more than one stock will fundamentally change our results, but the experiment seems worthwhile.

2. Impact of wealth. As the agents get more wealthy, they don't actually have more influence in the market under our CARA assumptions. There are several ways to change those assumptions.

3. Improved prediction. There are many ways to improve our agents' prediction methods. We have experimented briefly with neural network predictors, but did not find significantly different results.

4. Transition details. The transition between the two behavior regimes deserves detailed study. Is it really a sharp transition, or is it gradual? What are the sizes of the basins of attraction of the two regimes? How do they scale with the number of agents, and with other parameters?

5. Information control. It is possible to give different agents different information sets and thus explore the effect of private information. We can also provide information that is released periodically to all agents, like news reports.

6. Strategic behavior. We can have a longer time horizon so that the agents can look further ahead, rather than just one period at a time. Then we need to allow the agents to have strategic behavior over multiple periods.

Acknowledgements

We thank Paul Tayler and Brandon Weber for their collaboration in this project. This work was supported in part by the Santa Fe Institute's Economics Program, including grants from Citicorp, Coopers and Lybrand, McKinsey and Company, the Russell Sage Foundation, and the Walker Foundation; and by core funding to the Santa Fe Institute from the John D. and Catherine T. MacArthur Foundation, the National Science foundation, and the U.S. Department of Energy; and by gifts and grants from individuals and members of the Institute's Business Network for Complex Systems research.

References

[1] W.B. Arthur, "On Learning and Adaptation in the Economy," Santa Fe Institute Paper 92–07–038, 1992.

[2] W.B. Arthur, J.H. Holland, B. LeBaron, R.G. Palmer, and P. Tayler, "Asset Pricing Under Endogenous Expectations in an Artificial Stock Market," pp. 15–44 in *The Economy as an Evolving Complex System II*, edited by W.B. Arthur, D. Lane, and S.N. Durlauf, Addison-Wesley, 1997.

[3] T. Bollerslev, R.Y. Chou, N. Jayaraman, and K.F. Kroner, "ARCH Modeling in Finance: A Review of the Theory and Empirical Evidence," *Journal of Econometrics*, Vol. 52, pp. 5–60, 1990.

[4] J.A. Frankel and K.A. Froot, "Chartists, Fundamentalists, and Trading in the Foreign Exchange Market," *AEA Papera and Proceedings*, Vol. 80, pp. 181–185, 1990.

[5] R.E. Lucas, "Asset Prices in an Exchange Economy," *Econometrica*, Vol. 46, pp. 1429–1445, 1978.

[6] R.G. Palmer. W.B. Arthur, J.H. Holland, B. LeBaron, and P. Tayler, "Artificial Economic Life: A Simple Model of a Stockmarket," *Physica D*, Vol. 75, pp. 264–274, 1994.

[7] A. Shleifer and L.H. Summers, "The Noise Trader Approach to Finance," *Journal of Economic Perspectives*, Vol. 4, pp. 19–33, 1990.