Martin-Luther-Universität Halle-Wittenberg

Wirtschaftswissenschaftliche Fakultät

The Santa Fe Artificial Stock Market Re-Examined — Suggested Corrections

Norman Ehrentreich

Betriebswirtschaftliche Diskussionsbeiträge

Beitrag Nr. 45/02

Große Steinstraße 73 – D-06099 Halle/Saale

The Santa Fe Artificial Stock Market Re-Examined — Suggested Corrections

Norman Ehrentreich*
Martin-Luther-Universität Halle-Wittenberg

September 2002

Abstract

This paper rectifies a design problem in the Santa Fe Artificial Stock Market Model. Due to a faulty mutation operator, the resulting bit distribution in the classifier system was systematically upwardly biased, thus suggesting increased levels of technical trading for smaller GA-invocation intervals. The corrected version partly supports the Marimon-Sargent-Hypothesis that adaptive classifier agents in an artificial stock market will always discover the homogeneous rational expectation equilibrium. While agents always find the correct solution of non-bit usage, analyzing the time series data still suggests the existence of two different regimes depending on learning speed. Finally, classifier systems and neural networks as data mining techniques in artificial stock markets are discussed.

JEL Classification: G12; G14; D83

Keywords: Learning; Asset Pricing; Financial Time Series; Genetic Algorithms; Classifier Systems

*University of Halle-Wittenberg, Department of Banking and Finance, Universitätsplatz 8–9, 06099 Halle/Saale (Germany), E-mail: ehrentreich@wiwi.uni-halle.de, phone: +49-345-552-3453. The market model was programmed using Java and the RePast-library. The author wishes to thank Ulrike Neyer, Lars Schiefner, Manfred Jäger, and Blake LeBaron for valuable comments and suggestions. All remaining errors are mine.

1 Introduction

In the last decade, the use of agent-based simulations of markets has gained more and more acceptance among social scientists. This methodology has first been heavily used by physical scientists who simulate complex systems of many interacting particles. In financial economics, such a 'particle' is represented by an investor who interacts with other investors. A major advantage is that these models allow the removal of many restrictive assumptions that are required by analytical models for tractability. For instance, all investors in such a model could be modeled as heterogenous with respect to their preferences, endowments, and trading strategies.

Among the numerous agent-based simulations of financial markets (e.g., Cont and Bouchaud 2000, Farmer 2000, Chen, Lux and Marchesi 2001), the Santa Fe Artificial Stock Market Model (SFI-ASM) is one of the pioneering models and thus probably the most well-known and best studied.[1] It has been developed since 1989 and has been described in various papers (Arthur et al. 1997, Palmer et al. 1989, LeBaron et al. 1999). One of its main results is the identification of a single parameter, i.e., the learning speed of agents, which is able to shift the model to either a regime that is close to the homogeneous rational expectation equilibrium, or to a more complex regime that better fits the empirical facts. The complex regime emerges for fast learning rates and exhibited technical trading that dominated fundamental strategies. Consequently, Joshi, Parker, and Bedau (1988 and 2002) concluded that financial markets inevitably are at sub-optimal equilibria, and that technical trading is caused by a typical prisoner's dilemma.

However, a closer investigation of the genetic algorithm that updates the trading rules of agents reveals a flaw that shifts the outcome towards these results. In particular, a faulty mutation operator causes a systematic upward bias in the level of set bits in the condition parts of trading rules, thus suggesting increased levels of fundamental and technical trading for faster learning speeds. Yet, when eliminating these technical influences on the bit-level by a corrected mutation operator and a rule consistency check, the genetic algorithm (GA) of the classifier system always discovers the correct homogeneous rational expectation equilibrium (hree) of non-bit usage, no matter

[1] There are, in fact, several 'generations' of the SFI-ASM on different programming platforms. An overview over the SFI market history can be found in LeBaron (2002) and Johnson (2002). A current objective-C version using the Swarm package is currently hosted by Paul Johnson at http://ArtStkMkt.sourceforge.net. The results reported in this paper are obtained with a newly programmed Java version using the RePast library. The source code will be soon made available on the internet.

which GA-invocation interval is used. Nonetheless, for slow learning speeds, the simulated time series still suggest the existence of a hree-regime, while in the fast learning case, a complex-regime with increased price volatility and trading volume emerges.

This paper indicates that it is much harder to generate technical trading and herding behavior in a classifier-based artificial stock market than initially thought. In the SFI-ASM, technical trading was supposed to emerge as a synchronization of individual trading rules by means of coordinating on identical condition parts. However, this paper suggests that the possibilities provided by the classifier system to coordinate on similar technical trading strategies are too weak. If one wishes to reproduce the empirical facts of technical trading and herding behavior, other mechanisms must be found to be used in agent-based models with classifier systems.

In order to be self-contained, Section 2 first introduces the basic structure of the original SFI-ASM with its main results. At the end of that section, the flawed mutation operator is analyzed and its effects on the bit distribution are illustrated. In Section 3, an updated mutation operator and a rule consistency check are developed such that any technical influence on the resulting bit level can be excluded. Finally, the results of the original and updated version of the SFI-ASM are compared, and a short discussion of classifier systems and neural networks as data mining techniques in artificial financial markets concludes the paper.

2 The original SFI-ASM

2.1 The Basic Structure

The SFI-ASM is inhabited by N traders who are all initially endowed with one unit of risky stock and $20,000$ units of cash. During each period, traders have to decide how much to invest in risky stock and how much to keep in cash assets which yields a risk-free rate of return r_f.

The stock pays a stochastic dividend per period which is generated by a mean-reverting autoregressive Ohrnstein-Uhlenbeck process

$$d_{t+1} = \bar{d} + \rho(d_t - \bar{d}) + \epsilon_{t+1}, \tag{1}$$

with $\epsilon_t \sim N(0, \sigma_\epsilon^2)$. Traders are homogeneous with respect to their utility function which is a myopic, constant absolute risk-aversion (CARA) expected

utility function

$$U(W_{t+1}) = -e^{-\lambda W_{t+1}}, \tag{2}$$

with λ being the degree of risk-aversion. Under the assumption of a normal distribution of returns, agents maximize their expected utility subject to the budget constraint

$$W_{i,t+1} = x_{i,t}(p_{t+1} + d_{t+1}) + (1 + r_f)(W_{i,t} - p_t x_{i,t}), \tag{3}$$

where $x_{i,t}$ is the amount of stock an agent i holds in period t. The optimal amount of stock $\widehat{x_{i,t}}$ that an agent desires to hold is then determined as

$$\widehat{x_{i,t}} = \frac{E_{i,t}[p_{t+1} + d_{t+1}] - p_t(1 + r)}{\lambda \sigma^2_{t,p+d}}, \tag{4}$$

where $E_{i,t}[p_{t+1} + d_{t+1}]$ is his expectation in t about the next period's realization of the stock's price and dividend, and $\sigma^2_{t,p+d}$ the empirically observed variance of the combined price plus dividend time series. A specialist collects all effective demands as well as its partial derivatives with respect to the price and tries to find a market clearing price in an iterative process. If complete market clearing is not reached after a specified number of trials, one side of the market will be rationed.

While traders are homogenous with respect to their utility functions and degrees of risk aversion, they have heterogeneous expectations about future prices and dividends $E_{i,t}[p_{t+1} + d_{t+1}]$. It could be interpreted that they differ in the way they process an identical information set. Forecasts are derived via trading rules from which each agent possesses an individual set of 100 rules. A rule consists of a condition part, a forecast part (predictor), its fitness value, and forecast accuracy. Thus, a forecast is derived according to

if (condition fulfilled) then (use predictor to derive forecast).

The condition parts are checked against a Boolean market descriptor D_t which holds current and past price and dividend information. For example, a particular market state could be that the price of the stock is greater than n-times its fundamental value, while at the same time, the 25-period moving average of the stock price is greater than the price mean. When a particular predefined condition is met, the corresponding descriptor bit is set to 1, otherwise to 0.

The condition part, on the other hand, is coded as a ternary string holding either 1 or 0, depending on whether the corresponding bit in the market

descriptor has to be matched or not, and # if the rule ignores that particular descriptor bit.[2] Rules with numerous #-signs are quite general and hence, they will be activated more often than more specific rules. The bits of a trading rule may be characterized as either technical or fundamental. Technical bits check only price or trading volume information, while fundamental bits relate the price of a stock to its fundamental value by using dividend information. For example, dividends and prices are checked to determine whether they increased or decreased, and whether they are above or below certain moving averages. Most importantly, prices are checked against a stock's fundamental value by comparing for each ratio in the brackets whether

$$\text{price x interest rate/dividend} > \left\{\frac{1}{2}, \frac{3}{4}, \frac{7}{8}, 1, \frac{9}{8}, \frac{5}{4}, \frac{3}{2}\right\} \tag{5}$$

is fulfilled. While all agents in the model may use fundamental bits, the fraction of agents that have access to technical bits is sometimes varied.

¿From the set of 100 individual trading rules that each agent possesses, normally more than one match the specified condition. All rules that fulfill this condition are marked as active, yet agents still have to choose one for their forecast production. This is done via the roulette wheel mechanism which favors rules with good fitness scores over those with low fitness values. Finally, a forecast is generated by a linear equation

$$E_{t,i}[p_{t+1} + d_{t+1}] = a_{i,j}(p_t + d_t) + b_{i,j} \tag{6}$$

with a_j and b_j being real valued parameters constituting the predictor part of a chosen trading rule j. Only in rare occasions, when no rules match the market descriptor, the parameters a and b are determined as a fitness weighted average of all a_j and b_j with $j = 1 \ldots 100$.

One period later, the accuracy of all activated rules is checked by comparing their predictions $E[p_{t+1} + d_{t+1}]$ with the actual realization of $(p_{t+1} + d_{t+1})$. A rule's forecast accuracy is determined as

$$\nu^2_{t,i,j} = \left(1 - \frac{1}{\theta}\right)\nu^2_{t-1,i,j} + \frac{1}{\theta}\Big[(p_t + d_t) - [a_{i,j}(p_{t-1} + d_{t-1}) + b_{i,j}]\Big]^2. \tag{7}$$

This forecast accuracy is measured as a weighted average of previous and current squared forecasting errors. The parameter θ determines the size of

[2]Technically, the units in the ternary strings should be called *trits*. A trit is the smallest unit that can hold three values. However, as is usually done in the literature, we will refer to them as bits.

the time window that agents take into account when estimating a rule's accuracy. As LeBaron et al. (1999, p. 1496) have pointed out, the value of θ is a crucial design question since it strongly affects the speed of accuracy adjustment and thus, learning in the artificial stock market. If $\theta = 1$, the rules would be judged only on the last period's performance and forecast accuracy would be strongly prone to noise. In the other extreme, however, as θ goes to ∞, agents would take all past information into account, implicitly assuming that they live in a stationary world. As in LeBaron et al (1999), a value of 75 is chosen for θ.

The forecast accuracy $\nu_{t,j}^2$ is used as a rule's variance estimate $\sigma_{t,(p+d)}^2$, which is used in equation (4). Furthermore, it is the main determinant of a rule's fitness

$$f_{t,j} = C - \left(\nu_{t,j}^2 + \texttt{bitCost} \times \texttt{specificity}\right), \tag{8}$$

with `specificity` being the number of conditions in a rule that are not ignored, `bitCost` as an associated cost for each bit set, and C as a positive constant to ensure positive fitness[3]. Attaching positive cost for every non-ignored bit could be interpreted as the cost of acquiring and evaluating new information. Penalizing rule specificity is also tantamount to a complexity aversion since it favors simple rules over more specific ones. Furthermore, it ensures that each checked condition contains useful information in the trading rule.

2.2 Learning and Rule Evolution

So far, agents have been equipped with a static rule set. Feedback learning in the stock market has taken place by identifying and using the rules that performed better than others, while the learning speed and quality were strongly dependent on the parameter θ. However, if an agent started with a rule set that contained only bad performing rules, in the absence of any other learning mechanism, she would not be able to find better ones. Thus, a genetic algorithm (GA) provides a way to alter the rule sets by replacing bad performing rules with new, possibly better ones[4]. By searching the possible search space in a random, yet not directionless fashion, the GA creates the basis for further explorative learning of the agents that takes place on a longer time scale than the accuracy estimation.

[3]Variable names as they appear in the source code of the model are typed in `courier`.

[4]Two useful introductions to genetic algorithms, which were originally developed by Holland (1975), are provided in Goldberg (1989) and Mitchell (1996).

For each agent, the GA is invoked every K periods on average and replaces the 20 worst rules of the rule set by newly created ones. In doing so, the GA uses the genetic operators of *mutation* and *crossover*. Mutation is an important part for any evolutionary algorithm and could be interpreted as learning by experiment or by unintended mistakes.[5] It helps to maintain a diverse population and avoids premature convergence of the search algorithm. For mutation, which is performed with a probability of 0.7 in the model, one parent is chosen by using tournament selection. Here, two candidates are randomly drawn from the rule set and the fitter one is selected to be the parent. A genetically identical offspring is created from the parent, and with a small mutation probability of 0.03, each bit in the condition part of the offspring is flipped at random. The real valued parameters of the predictor are changed by adding random numbers to them. The offspring's forecast accuracy is set at the median accuracy of all rules.

Contrary to mutation, crossover is a sexual genetic operator that requires two fit parents in order to work. Even though there are a variety of different crossover operators available, the original SFI model exclusively uses uniform crossover for the condition parts. Here, an offspring's bit is chosen with equal probability from the corresponding bit positions from either one or the other parent. Note that the fraction of bits set in the offspring is an unweighted average of the two parents' bit fraction. Thus, there is no systematic influence on average specificity through the working of the crossover operator.

parent 1	#	0	1	#	#	#	#	1	1
parent 2	#	0	1	1	1	0	0	0	0
offspring	#	0	1	#	1	#	0	1	0

Table 1: Example of uniform crossover on the condition part.

As for the real valued parameters, LeBaron et al. (1999, p. 1498) point out that there is little experience in the GA community regarding how to perform the crossover. Their approach is to construct the new parameter values by determining a weighted average of the two parents values, with $1/\sigma^2_{j,p+d}$ as the weight for each parent. The weights are normalized to sum up to 1.

[5]See, for instance, Riechmann (2001: 1021), or Dawid (1999: 68).

2.3 Experimental Results by the SFI-ASM

Depending on the GA-invocation interval, the original SFI-ASM was able to generate two different regimes (e.g., Arthur et al. 1997 and LeBaron et al. 1999). The so-called rational expectations regime emerged when agents had a slow learning rate, i.e., the GA was only seldom invoked, on average every 100 periods. Bit usage remained low and the agent's forecast parameters converged to their homogeneous rational expectation equilibrium (hree)-values, thus indicating that agents became more and more homogeneous.

On the other hand, the rich psychological or complex regime arose when agents had a fast exploration rate, i.e., the GA was often invoked. Here, the continuously co-evolving agents remained heterogeneous with respect to their bit usage and forecast parameters. In fact, the increased use of technical trading bits was often considered to be the most striking difference between the two regimes and was interpreted as an emergent property of the market. Furthermore, the price series exhibited unstable behavior such as bubbles and crashes, as well as other statistical properties like fat tails in the return distribution that can also be observed in real financial markets. Trading volume exhibited GARCH-behavior and was auto-correlated while having a positive cross-correlation with volatility and squared returns. Price volatility and risk premiums were significantly higher compared to the slow learning case. Since none of these nonlinear effects can be attributed to the underlying dividend process, they are an emergent property of the market process, i.e., the interactions of many heterogeneous agents. Thus, the SFI-model also supports the "interacting agent hypothesis" as proposed by Lux (1998) and Lux and Marchesi (1999).

By using a game theoretic framework, Joshi, Parker, and Bedau (2002) determined an agent's optimal learning rate in the SFI-ASM by analyzing the wealth that different types of agents have acquired. They found a unique strategic Nash equilibrium which clearly falls into the complex market regime, which is inefficient compared to other forecast revision rates. Thus, they concluded that financial markets can result in a sub-optimal equilibria. In their 1998 paper, Joshi, Parker, and Bedau concluded that technical trading emerges as a result of a typical prisoner's dilemma. Since technical traders can exploit short term patterns in the price series, technical trading generates more wealth compared to pure fundamental trading and is thus a dominant strategy for them. Since the market constitutes a symmetric simultaneous-move N-person game, a dominant strategy for one player is a dominant strategy for all players. In the symmetric Nash equilibrium, all rational traders are technical traders, yet this equilibrium is sub-optimal compared with a

hypothetical equilibrium in which traders have no access to technical trading rules.

2.4 The Problem: A Faulty Mutation Operator

Even though these results have a huge theoretical appeal, at least some of them have to be reconsidered in the light of the faulty mutation operator that was used in the SFI-ASM. This problem in the mutation operator might have systematically twisted the simulated outcomes towards the results as described above. Unlike crossover, the mutation operator is not neutral to the initial level of bits set and usually introduces an upward bias in the resulting bit level. Thus, any interpretation that is directly or indirectly linked to rule specificity and emergent technical trading must be treated with caution. In the fast learning regime, since the genetic algorithm, and hence the mutation operator, are invoked more often, the resulting final bit level is directly upwardly biased. A higher average rule specificity is, in turn, connected to a smaller fraction of activated rules per agent which might introduce instabilities and inefficiencies in this agent's ordering behavior.

In order to demonstrate the bit increasing effect through mutation, we must look at the bit transition probabilities as given in LeBaron et al. (1999: 1498). Once a bit is chosen for mutation with probability π, it is changed according to the bit transition probabilities as shown in table (2).

0	$\overset{0}{\longmapsto}$	0	0	$\overset{1/3}{\longmapsto}$	1	0	$\overset{2/3}{\longmapsto}$	#
1	$\overset{1/3}{\longmapsto}$	0	1	$\overset{0}{\longmapsto}$	1	1	$\overset{2/3}{\longmapsto}$	#
#	$\overset{1/3}{\longmapsto}$	0	#	$\overset{1/3}{\longmapsto}$	1	#	$\overset{1/3}{\longmapsto}$	#

Table 2: Transition probabilities in the original SFI-mutator.

LeBaron et al. assert that these transition probabilities would, on average, maintain the specificity, i.e., the fraction of #'s in a rule. However, if we denote the initial fraction of bits set before mutation with $P \in [0, 1]$, we find that the non-# bits are mutated to non-# bits with a probability of $P'_{|0,1 \mapsto 1,0} = \frac{1}{3}P$, while the probability that a #-bit is mutated to either 1 or 0 equals $P'_{|\# \mapsto 0,1} = \frac{2}{3}(1 - P)$. Thus, for any given P, the fraction of bits set after mutation is determined by adding the two probabilities above, i.e.,

$$P'_{|0,1,\# \mapsto 1,0} = \frac{1}{3}(2 - P). \tag{9}$$

Since $P'_{|0,1,\#\mapsto 1,0} \in [0,1]$ is a continuous function for all $P \in [0,1]$, we know by a fixed point theorem that an equilibrium exists. By repeatedly invoking the mutation operator, equation (9) converges to its equilibrium value of one half.[6]

Because the model usually functions well below this level, the mutation operator introduces an upward bias in the bit distribution. In other words, the mutation operator is not neutral to the initial level of non-# bits. This upward tendency is illustrated in figure (1) by varying the probability with which mutation is performed in the model.[7]

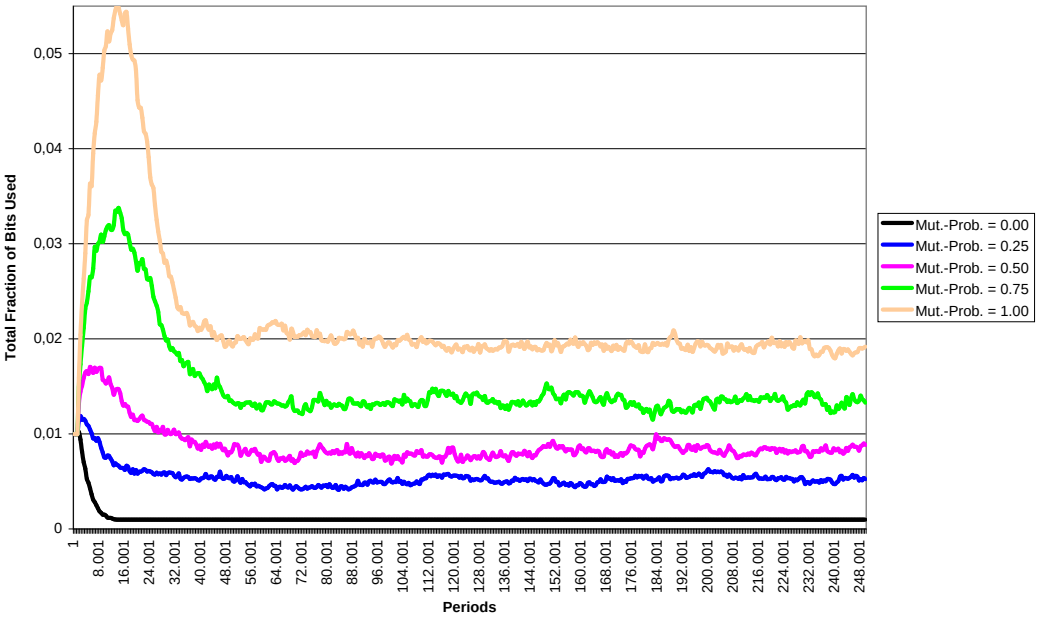

Figure 1: Total Fraction of Bits Used as a Function of Crossover Probability. Data were obtained from a cross section of 5 separate runs at different random seeds.

Even though the bit increasing effect of mutation is obvious, the theoretical equilibrium level of 0.5 is far from being attained. This is due to a variety of

[6]This result also holds if every bit in the bitstring is mutated with a probability of less than one. In the SFI-ASM, this mutation probability π is set to 0.03. Deriving the fraction of non-# bits in the same manner as above yields $P'_{|0,1,\#\mapsto 1,0} = \frac{2}{3}\pi(1-P) + P\left(1-\frac{2}{3}\pi\right)$. It is easy to check this formula by setting $\pi = 1$, which will yield equation (9), or by setting $\pi = 0$, which will yield $P'_{|0,1,\#\mapsto 1,0} = P$ since no mutation will ever be performed. The equilibrium level equals $1/2$, even though convergence occurs a little bit slower than before.

[7]The GA calls either the crossover operator with a probability of Π_{Co}, or the mutation operator with a probability of $\Pi_M = 1 - \Pi_{Co}$.

other model parameters that also influence the final bit distribution, usually in the opposite direction. First of all, the GA replaces only a fraction of the rules, typically one fifth, by newly created ones. Secondly, by looking at equation (5), we realize that the GA might produce illogical rules such as rule 1 in table (4). A typical remedy is to work with a larger rule set and to invoke a generalization procedure for any rule that has not been matched for the last `maxNonActive` periods. This procedure lowers rule specificity by converting set bits to # with probability `genFrac`. Third, the `bitCost` parameter penalizes every non-# bit and thus, the GA preferably selects rules with below average specificity for mutation and crossover. Hence, positive `bitCost` cause a downward tendency in the final bit level.

Since positive bit cost causes the resulting bit fraction to be attracted towards the zero bit level, LeBaron et al. (1999) view it as a necessary robustness check to see whether the artificial stock market can withstand the inflow of technical trading rules. The extensive use of trading bits in the fast learning regime compared to the hree-regime leads them to conclude that they reflect emergent technical trading which must be an expression of some fitness-based advantages over more general rules.

However, since we know by now that the mutation operator has a fixed point of 0.5 towards which the equilibrium bit level is torn, we cannot assert simply by the existence of technical trading bits that technical trading has emerged. Furthermore, the presence of trading bits does not necessarily mean that agents use them. To draw such a conclusion, one would actually have to count how often general or more specific rules were selected for use by the agents.

3 A Corrected Version of the SFI-ASM

3.1 Suggested Corrections

In order to obtain valid results with respect to the resulting bit levels, one should only implement bit neutral operators and procedures. While the bit decreasing effect of `bitCost` is desirable as it is a fitness-based influence, it is not desirable for the generalization procedure and the mutation operator since it is completely technical and economically not interpretable.

Thus, the corrected version of the model to which I will henceforth refer to as the NESFI-ASM (Norman Ehrentreich's SFI-ASM), is equipped with a bit neutral mutation operator and a rule consistency check which ensures

that only logical rules are created by the GA.[8]

Unlike the original SFI-ASM, the bit neutral mutation operator works with dynamically adjusting bit transition probabilities. To allow for diverging fundamental and technical bit usage in the model, we distinguish between the initial fraction of fundamental bits set $F_{fund.}$, and the initial fraction of technical bits set $F_{techn.}$. Thus, the bit neutral mutation operator is characterized by the bit transition probabilities shown in table (3)[9].

fundamental bits	0	$\overset{0}{\longmapsto}$	0	0	$\overset{F_{fund.}}{\longmapsto}$	1	0	$\overset{1-F_{fund.}}{\longmapsto}$	#
technical bits	0	$\overset{0}{\longmapsto}$	0	0	$\overset{F_{techn.}}{\longmapsto}$	1	0	$\overset{1-F_{techn.}}{\longmapsto}$	#
fundamental bits	1	$\overset{F_{fund.}}{\longmapsto}$	0	1	$\overset{0}{\longmapsto}$	1	1	$\overset{1-F_{fund.}}{\longmapsto}$	#
technical bits	1	$\overset{F_{techn.}}{\longmapsto}$	0	1	$\overset{0}{\longmapsto}$	1	1	$\overset{1-F_{techn.}}{\longmapsto}$	#
fundamental bits	#	$\overset{\frac{1}{2}F_{fund.}}{\longmapsto}$	0	#	$\overset{\frac{1}{2}F_{fund.}}{\longmapsto}$	1	#	$\overset{1-F_{fund.}}{\longmapsto}$	#
technical bits	#	$\overset{\frac{1}{2}F_{techn.}}{\longmapsto}$	0	#	$\overset{\frac{1}{2}F_{techn.}}{\longmapsto}$	1	#	$\overset{1-F_{techn.}}{\longmapsto}$	#

Table 3: Transition Probabilities in a Bit-Neutral Mutation Operator.

However, since $F_{fund.}$ and $F_{techn.}$ are averaged over all rules of an agent, this mutation operator is still not completely freed from an upward pressure in the output fraction of bits set. Since specificity bears some costs by lowering a rule's fitness, general rules tend to be selected for mutation, provided that their predictive power is at least as good as those of more specific rules. Thus, the fraction of bits set in a single rule tends to be augmented by mutation, not lowered, since the average fraction of bits set in an agent's rule set is, in general, bigger than the fraction of bits sets in those rules that were chosen. However, the magnitude of that effect is probably negligible.

The consistency check, on the other hand, corrects inconsistencies produced by the GA. It does so by expressing a logical bit sequence with the least amount of non-# bits necessary, i.e., one or two. However, depending on whether the checking procedure starts analyzing the bit sequence from

[8]Even though the need for a generalization procedure is drastically reduced by the consistency check, some bit combinations of the condition part might never occur, and thus, the generalization procedure is still kept active.

[9]Since the main purpose of mutation is usually to maintain a diverse population by arbitrarily introducing new bits, this function might be hampered by the suggested transition probabilities. Once a zero fraction of used bits is reached, there would be no way to reintroduce condition bits into the rule set. Thus, ensuring strictly positive transition probabilities such as $\max[F_{fund.}, minProb]$ with $minProb > 0$ might be appropriate.

the lower bits upward or vice versa, different results will be returned by it. Hence, both variants will be chosen with equal probability. Some examples of input and output sequences are given in table (4).

$\frac{\text{price x interest Rate}}{\text{dividend}} >$	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{7}{8}$	1	$\frac{9}{8}$	$\frac{5}{4}$	$\frac{3}{2}$
rule 1	#	#	0	1	#	0	#
rule 2	#	#	1	1	#	0	#
$\overleftarrow{\text{rule 1}} \longmapsto$ rule 3	#	#	#	1	#	0	#
$\overrightarrow{\text{rule 1}} \longmapsto$ rule 4	#	#	0	#	#	#	#

Table 4: Creating Logical Rules through Consistency Check.

The first rule contains a contradictory condition. Rule 2, on the other hand, might be considered a corrected version. However, the third rule demonstrates that the same information can be coded by using less activated bits. This rule would be returned by the consistency check if it starts analyzing from the higher order bits, otherwise the fourth rule would be returned.

After having eliminated all technical influences on the bit distribution, one can be sure that every single bit in a trading rule emerges through competition and fitness considerations.[10] Chances are high that bits will only survive if they indeed contain useful information.

3.2 Experimental Results

3.2.1 Agents Forecast Properties

As one could expect, an analysis of the real valued forecasting parameters as well as the mean variance of all rules does not exhibit substantial differences from the original SFI model. Thus, the differences for these parameters in the slow and fast learning regime must be caused by other reasons than a more or less extensive use of condition bits.

However, when comparing the bit usage between the original SFI-ASM and the NESFI-ASM, one immediately recognizes substantial differences. For

[10] A pleasant side effect of the consistency check is that one can work with a smaller rule set, since every rule could be principally activated. The saved computation time is probably larger than the time needed for the checking procedure.

all GA-intervals, no emergent technical trading can be detected, not even temporarily. More interestingly, agents completely abandon technical and fundamental bit usage in the long run and do not check any conditions at all.[11] This result is so robust that one could doubt the proper working of the classifier system. Thus, the model behavior was tested for classifier mode and non-classifier mode. For the latter, agents had no access to condition bits at all. In both cases, they were confronted with a periodic square wave dividend stream. Even though the simulated price series tracked the crude risk neutral price astoundingly well in the non-classifier mode, the tracking behavior in classifier mode was far superior for most GA intervals. Agents started to use some fundamental as well as technical bits extensively while neglecting others, and were thus able to predict prices much more accurately. Consequently, the classifier agents also acquired more wealth than the non-classifier agents.

Hence, it is shown that the classifier system works very efficiently. When confronted with periodic dividend data, it detects these patterns, yet when working with stochastic data, it also discovers the "right" solution of non-bit usage. Even though the mean-reverting dividend process is able to produce short term trends toward its mean, these are by no means regular. Thus, in the long run, the stochastic nature of the dividend process dominates any (random) short term trends and pattern.

However, depending on the GA-invocation interval, it may sometimes be relatively difficult for the GA to discover the non-bit usage solution. In figure (2), the average number of GA-invocations needed to reach the zero-bit level is plotted as a function of the invocation interval. Here, one realizes a strong increase for the fundamental bits for interval lengths between 10 and 250 periods.[12] The GA probably temporarily detects what it believes to be short term patterns, yet in the long run, when testing and acting upon these adapted rules, it realizes that they do no better than the all-# rule. It is also apparent that irrespective of the invocation interval length, technical bits are driven out of the rule set equally fast.[13]

[11] In order to better demonstrate the strong attraction of the zero bit level, a positive minimum transition probability for the mutation operator has not been implemented.

[12] The picture is quite different if the two parameters $F_{techn.}$ and $F_{fund.}$ are global for all agents. There, a significant amount of social learning occurs such that the maximum number of GA-invocations needed to find the non-bit solution does not exceed 45.

[13] There is a small possibility that one has simply not found the "right" conditions to code. The neural network approach by LeBaron (2001b) endogenizes the required hard wiring of key breakpoints in the classifier system, yet when altering some of the price ratios or moving averages break points, or including other technical information such as trading volume, the qualitative behavior did not change.

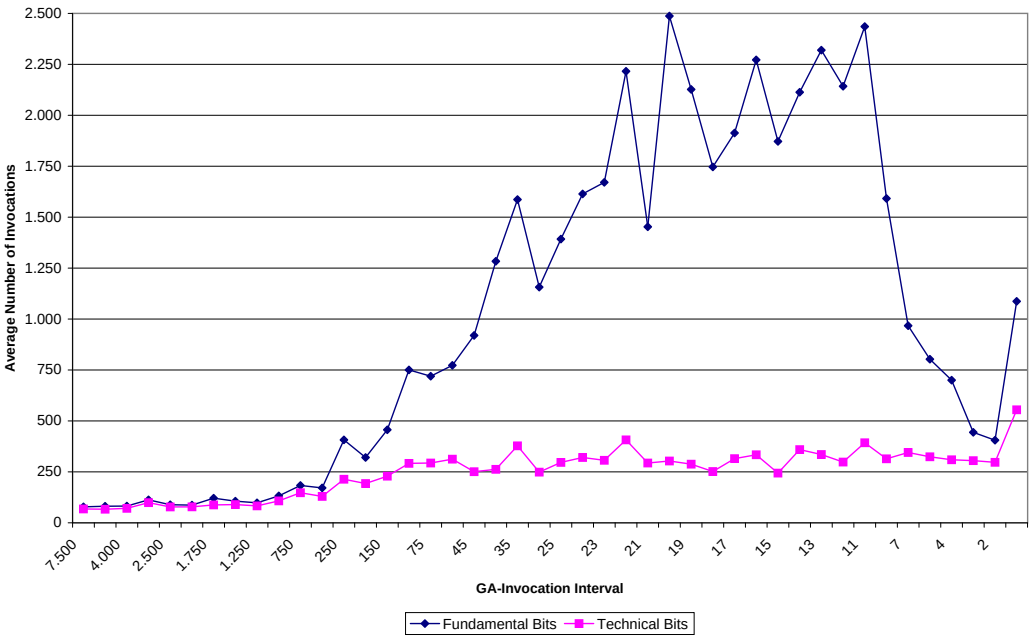

Figure 2: Average Number of GA-Invocations Needed Until Zero-Bit Level is Reached (averaged over 25 runs)

We will refer to the bit level behavior as part one of the *Marimon-Sargent-Hypothesis*. According to Waldrop (1992: 270), Ramon Marimon and Thomas Sargent claimed in discussions with John Holland and Brian W. Arthur that adaptive classifier agents in an artificial stock market model would quickly learn the neoclassical rational equilibrium solution. They were led to this statement through their own research (Marimon, McGrattan and Sargent: 1990) in which they assigned adaptive classifier agents to solve Wicksell's triangle in a Kiyotaki-Wright (1989) type model. There, they found that the agents always discovered the neoclassical solution, i.e., the good with the lowest storage cost emerged as a medium of exchange.

Unlike the original SFI model, the corrected NESFI-ASM finally supports the Marimon-Sargent-Hypothesis. All agents realize that, under the given dividend process, all they need for their forecast production is the last period's price and dividend information, which is compatible with the linear rational expectation equilibrium.

3.2.2 Time Series Properties

A typical neoclassical rational equilibrium solution would not only be characterized by a total neglect of any additional information contained in condition

bits, but it would furthermore exhibit "nice" price series properties. Whether the time series are consistent with the neoclassical solution will be considered as part two of the Marimon-Sargent Hypothesis.

One should keep in mind that the proposed changes to the GA only affect the condition part and not the real valued forecast parameters of a trading rule. Thus, one would expect the two models to produce similar time series, i.e., "well behaved" ones for the slow learning case and more complicated ones for faster GA-invocation intervals.

This hypothesis was tested by running the same statistical tests on the time series and comparing the results with those published in LeBaron et al. (1999: 1501). All model parameters were set to the same values reported there. The hree-case serves as a benchmark in which the dividend and market price should be a linear function of their first order lags. Thus, they are regressed on a lag and a constant

$$p_{t+1} + d_{t+1} = a(p_t + d_t) + b + \epsilon_t, \tag{10}$$

with the residuals ϵ being i.i.d. and N(0,4) distributed. The results are summarized in table (5).

First of all, one notices that the corrected NESFI-version produces time series that are generally closer to the hree-benchmark. Most statistics for the NESFI-version show smaller deviations from the hree-benchmark than the corresponding SFI-values. The standard deviations in the residuals are smaller, thus indicating less price variability. Excess kurtosis is almost negligible for both the fast and slow learning cases, which contradicts the empirical fact of fat-tailed return distributions. Yet, when further enhancing the learning speed, both the increase in standard deviation and excess kurtosis suggest that the NESFI-model shifts into a complex regime for faster learning rates than the original SFI-model. This corresponds closely with the fact that the GA had increasing difficulties finding the correct solution of non-bit usage in the condition parts for invocation intervals less than 100. The autocorrelation in the residuals, as shown in the third row, demonstrates that there is little linear structure remaining. As LeBaron et al. indicate, any artificial stock market should exhibit negligible autocorrelations since they are very low for real markets. The next row reports the means of the test statistics for the ARCH test proposed by Engle (1982). There is considerably less ARCH dependence in the residuals for the NESFI-version. It is interesting to note that even for very small GA invocation intervals, some test runs are not able to reject the no-ARCH hypothesis. Only for a GA invocation in

Description	GA 1000		GA 250		GA 20	GA 1
	NESFI	SFI	NESFI	SFI		
Std. Dev.	2.084	2.135	2.141	2.147	2.229	3.397
	(.009)	(.008)	(.013)	(.017)	(.013)	(.034)
Excess kurtosis	0.004	0.072	0.001	0.320	0.050	9.046
	(.009)	(.012)	(.001)	(.020)	(.011)	(1.56)
ρ_1	0.011	0.036	0.014	0.007	0.029	0.491
	(.002)	(.002)	(.002)	(.004)	(.001)	(.006)
ARCH(1)	2.610	3.159	2.754	36.98	5.722	1871.9
	[0.20]	[0.44]	[0.40]	[1.00]	[0.48]	[1.00]
ρ_1^2	0.013	0.017	0.015	0.064	0.020	0.425
	(.002)	(.002)	(.004)	(.004)	(.003)	(.017)
BDS	1.06	1.28	1.10	3.11	1.44	38.63
	[0.20]	[0.24]	[0.24]	[0.84]	[0.28]	[1.00]
Excess return	1.52%	2.89%	1.59%	3.06%	1.51%	25.34%
	(.02%)	(.03%)	(.03%)	(.05%)	(.03%)	(3.41%)
Trading volume	0.244	0.355	0.271	0.706	0.876	1.359
	(.008)	(.021)	(.007)	(.047)	(.009)	(.015)

Table 5: Comparison of the NESFI and SFI-version of the model. Means over 25 runs. Numbers in parentheses are standard errors estimated using the 25 runs. Numbers in brackets are the fraction of tests rejecting the no-ARCH or iid-hypothesis for the ARCH and BDS tests, respectively, at the 95% confidence level.

every period, can extreme ARCH-behavior for all test runs be observed.[14] In row five, the first order autocorrelation of the squared residuals is another test for volatility persistence. Again, it increases for faster learning speed, but is generally lower than for the SFI-case.

The BDS test in row six is a test for nonlinear dependence developed by Brock et al. (1996). Its test statistic is asymptotically standard normally distributed under the null hypothesis of independence. There are two free parameters for this test. The distance r is measured as a fraction of the standard deviation and has been set to a value of 0.5, while for the embedding dimension m, a value of two is chosen. One can notice an increasing amount of nonlinearities for faster exploration rates, yet again, it is substantially lower for the NESFI-version. Since this test usually rejects the hypothesis of independence for most financial time series, the NESFI-results indicate

[14]Even for an invocation interval of two, the no-ARCH hypothesis cannot be rejected for 16% of the test runs.

that financial markets operate at a learning speed that is too fast. Trading volume, which should be zero in the hree-case, increases significantly for faster learning speeds. This points to a greater degree of heterogeneity between the agents.

Overall, the original conclusion that the learning speed affects the outcome towards a hree-regime and complex regime can still be confirmed after the proposed change. However, complex time series behavior seems to emerge only for substantially faster learning speeds than in the SFI-version. It is also apparent that the NESFI-results are generally closer to the hree-benchmark. An explanation for this behavior could be the larger pool of activated trading rules from which agents can now choose. In the SFI-version, the increase in the bit level caused substantially fewer rules to be activated. If, on average, only a few general rules per agent were activated, agents repeatedly acted upon them regardless of their predictive power. If the forecast parameters of these rules were to diverge from the hree-values, less efficient time series were likely to occur.[15] In other words, the larger the pool of activated trading to choose from, the larger the probability of choosing a good one.

3.2.3 Wealth Behavior

Joshi, Parker, and Bedau (2002) derived the optimal revision rate of investors via a game theoretic analysis of wealth levels. There, they found that faster learning agents perform better than slow learning agents by exploiting short term trends and pattern that the slow learners discard as non-profitable in the long run. They computed a symmetric Nash-equilibrium for a GA-invocation interval of 100, which was in the complex regime with technical trading. Due to lower general wealth levels in the economy compared to the hree-case, they concluded that financial market equilibria would inevitably be sub-optimal.

Unfortunately, these results can only be partly replicated with the NESFI-version. Differences in wealth levels due to varying invocation intervals emerge only in the first periods of the market in which the equilibrium has not yet been found. The faster the GA-invocation interval for an agent, the faster she will learn the correct risk-adjusted hree-price of the stock. If all other agents have longer revision rates, the fast learning agent is able to ex-

[15] This problem could have been fixed in the SFI-version by imposing a minimum number of activated rules per agent from which to choose. If fewer rules are activated, agents would resort to the Select Average mechanism, i.e., using forecast parameters that are a fitness-weighted average of all rules. In the SFI-version, this minimum number was effectively fixed to one activated rule per agent.

ploit slower learning agents as long as the market price has not come close to the equilibrium level. Once this has happened, wealth levels rise almost proportionally for fast and slow learners, yet on a different level. In addition, classifier agents do considerably better in the warm-up phase of the market than non-classifier agents, yet in the long-run, the classifier-agents mutate to non-classifier agents, thereby leaving no opportunity for further wealth-divergence.

Although weaker, a similar temporary effect on wealth levels can be found for technical and fundamental traders. In the NESFI-version, traders that have access to technical trading bits do slightly better than pure fundamental traders, but only in the beginning. Thus, the conclusion by Joshi, Parker, and Bedau (1998) that technical trading emerges due to a prisoner's dilemma, and that rational traders are technical traders, cannot be upheld by the NESFI-ASM in the long run. This conclusion was also probably caused by temporary warm-up effects.

Finally, the NESFI-agents with the corrected mutation operator were competing against old SFI-agents. Since the abandonment of the classifier system in the NESFI-version was totally fitness-driven, one expected no differences. This hypothesis could be totally verified. For all phases of the market, none of the groups did significantly better than the other.

4 Summary and Conclusion

After having remedied a technical problem of the SFI-ASM, the corrected version supports the Marimon-Sargent-Hypothesis that agents in an artificial stock market will usually find the hree-solution. Independent of learning speed as modelled by the GA-invocation interval, agents realize that any temporary pattern in the time series they detect are random, and not worth acting upon in the long run. Thus, they neglect any additional information provided by the classifier system. The latter works so efficiently for the given dividend process that it finally reduces itself to absurdity. It seems that a classifier system is not well-suited for generating positive feedback trading or self-fulfilling prophecies.[16] Thus, if we are only interested in the long run equilibrium behavior of this artificial stock market, one could tremendously

[16]It might be possible to introduce technical trading if agents would be allowed to form coalitions, i.e., if they could coordinate on certain positive-feedback strategies. While individual agents can temporarily use positive feedback strategies, due to non-synchronization of trading strategies between agents, they might not have the critical mass to produce a regular pattern that will not vanish over time.

simplify it by abandoning the classifier system altogether.[17]

However, it is obvious that at least some real world traders use technical trading tools and try to discover regularities in the price series that can be exploited. Thus, the question remains whether we want our agents to analyze past time series data with some kind of data mining technique. Given the empirical facts of real financial data of clustered volatility, fat tails, crosscorrelation of returns and volume, and other random walk deviations, the use of some data mining technique in an artificial market becomes desirable since it opens up possibilities for the agents to deviate from the hree-solution. Yet, we have to be aware that any technique that indeed deviates from the hree-solution in an artificial stock market model would obviously be less efficient than the classifier system which always seems to find the correct solution in the current setting.

However, classifier systems are regarded more and more as a "less than perfect tool for modern research on financial markets" since they are accompanied by extensive costs such as the hard wiring of key breakpoints (LeBaron 2002). Neural networks have thus become popular not only in real financial markets, but recently, also in artificial stock markets (Beltratti et al. 1996, LeBaron 2001b). However, Davis (1989a, p. 375) regards classifier systems and neural networks as functionally equivalent. He shows that any classifier system can be transformed into an isomorphic neural network (Davis 1989b) and vice versa (Davis 1989a). Thus, a properly specified neural network should find the same solution as a classifier system, given that both are provided with the same information set. Since a classifier system is more well-suited to economic interpretation than the neurons and weights of a neural network, it appears to have advantages over the latter approach in an artificial stock market model. Thus, in my assessment, the controversy between classifier systems and neural networks has not yet been decided and further research in comparing both approaches seems necessary.

Regardless of which technique is finally used, when massively extending the information set, it will become more and more difficult to filter out noise. Given the almost infinite possibilities in the real world to analyze useful (and useless) data, some regularities will always be found. Whether they will persist in the long run cannot yet be told. Thus, it is possible that investors in real financial markets are simply too impatient to discard non-profitable strategies or lack the necessary memory to do so. In a way, the constant departure of experienced traders and arrival of new traders might induce

[17]This is true only for stochastic dividends. As soon as periodic components are introduced, classifier agents perform better than non-classifier agents.

financial markets to constantly function out of their long run equilibrium. Given the result in figure (2) and the average lifetime of an investor, any practical learning speed could put us in the region where it is extremely difficult to discover the hree-solution.

Due to the apparent "inability" of the NESFI-ASM to generate technical trading, further research should identify conditions under which this is possible. I suspect that it will not be linked to the kind of data mining system being used; more attention should be given to agent interaction. In the NESFI-ASM, these interactions were only indirect through price effects and seemed insufficient for generating positive feedback trading. Yet, if agents were allowed to communicate and collaborate by forming coalitions, they could coordinate on certain trading strategies, making them a self-fulfilling prophecy. Thus, studying the differences between individual learning as in the NESFI-ASM and social learning might be a more promising way to gain insight into the working of real financial markets.

5 References

Arthur, W.B., Holland, J., LeBaron, B., Palmer, R., Tayler, P., 1997. *Asset pricing under endogenous expectations in an artificial stock market.* in: Arthur, W.B., Durlauf, S., Lane, D. (Eds.), The Economy as an Evolving Complex System II. Addison-Wesley, Reading, MA., pp. 15–44.

Beltratti, A., Margarita, S., Terna, P., 1996. *Neural Networks for Economic and Financial Modeling.* International Thomson Computer Press, London.

Brock, W. A., Dechert, W. D., Scheinkman, J. A., LeBaron, B., 1996. *A test for independence based on the correlation dimension.* Econometric Reviews **15** (3), 197–235.

Chen, S.H., Lux, T., Marchesi, M., 2001. *Testing for non-linear structure in an artificial financial market.* Journal of Economic Behavior and Organization **46** (3), 327-342.

Cont, R., Bouchaud, J.-P., 2000. *Herd behavior and aggregate fluctuations in financial markets.* Macroeconomic Dynamics **4** (2), 170–196.

Dawid, H., 1999. *Adaptive Learning by Genetic Algorithms: Analytical Results and Applications to Economic Models.*, 2nd., rev. and enlarged edition, Berlin et al., Springer.

Davis, L., 1989a. *Mapping neural networks into classifier systems*, in: Schaffer, J.D. (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA., pp. 375–378.

Davis, L., 1989b. *Mapping classifier systems into neural networks*, in: Touretsky, D.S. (Ed.), *Advances in Neural Information Processing 1*, Morgan Kaufmann, San Mateo, CA., pp. 49–56.

Engle, R.F., 1982. *Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation.* Econometrica 50, 987–1007.

Farmer, J.D., 2000. *A simple model for the nonequilibrium dynamics and evolution of a financial market.* International Journal of Theoretical and Applied Finance **3** (3), 425-441.

Goldberg, D.E. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading, MA.

Holland, J.H., 1975. *Adaptation in Natural and Artificial Systems.* Ann Arbor, University of Michigan Press, (reprinted, MIT Press, 1992).

Johnson, P.E., 2002. *Agent-Based Modeling: What I Learned from the Artificial Stock Market.* Social Science Computer Review, **20** (2), 174–186.

Joshi, S., Parker, J., Bedau, M.A., 1998. *Technical trading creates a prisoner's dilemma: Results from an agent-based model.* Santa Fe Institute Working Paper 98–12–115.

Joshi, S., Parker, J., Bedau, M.A., 2002. *Financial markets can be at sub-optimal equilibria.* Computational Economics **19** (1), 5–23.
http://ipsapp002.lwwonline.com/content/search/4624/23/1/fulltext.pdf

Kiyotaki, N., Wright, R., 1989. *On money as a medium of exchange.* Journal of Political Economy **97** (4), 927–954.

LeBaron, B., Arthur, B.W., Palmer, R., 1999. *Time series properties of an artificial stock market.* Journal of Economic Dynamics and Control **23** (9–10), 1487–1516.

LeBaron, B., 2001b. *Empirical regularities from interacting long- and short-memory investors in an agent-based stock market.* IEEE Transactions on Evolutionary Computation **5** (5), 442–455.

LeBaron, B., 2002. *Building the Santa Fe Artificial Stock Market.* working paper, http://people.brandeis.edu/ blebaron/wps/sfisum.pdf

Lux, T., 1998. *The socio-economic dynamics of speculative markets: interacting agents, chaos, and the fat tails of return distributions.* Journal of Economic Behavior and Organization **33** (2), 143–165.

Lux, T., Marchesi, M., 1999. *Scaling and criticality in a stochastic multi-agent model of a financial market.* Nature 397, 498–500.

Marimon, R., McGrattan, E., Sargent, T., 1990. *Money as a medium of exchange in an economy with artificially intelligent agents.* Journal of Economic Dynamics and Control **14** (2), 329-373.

Mitchell, M., 1996. *An Introduction to Genetic Algorithms.* MIT Press, Cambridge, MA.

Riechmann, T., 2001. *Genetic algorithm learning and evolutionary games.* Journal of Economic Dynamics and Control **25** (6–7), 1019–1037.

Waldrop, M.M., 1992. *Complexity: The Emerging Science at the Edge of Order and Chaos.* New York: Touchstone.