A Corrected Version of the Santa Fe Institute Artificial Stock Market Model

Norman Ehrentreich*
Martin-Luther-Universität Halle-Wittenberg

December 23, 2004

Abstract

This paper rectifies a design problem in the Santa Fe Institute Artificial Stock Market Model. The mutation operator caused the resulting bit distribution to be systematically upwardly biased, thus suggesting emergent technical trading at faster learning speeds. The modified version now partly supports the Marimon-Sargent-Hypothesis which states that adaptive classifier agents in an artificial stock market will discover the homogeneous rational expectation equilibrium. While agents always learn the correct solution of non-bit usage, analyzing the simulated price series reveals that the updated model still shifts into a more complex regime, however, only at faster learning rates than the original model suggests.

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, Heinz-Peter Galler, Blake LeBaron and two anonymous referees 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 could be modeled as heterogeneous with respect to their preferences, endowments, and trading strategies.

Among the numerous agent-based simulations of financial markets [e.g., Levy, Levy, and Solomon (1994), Lux and Marchesi (1999), or Cont and Bouchaud (2000)], the Santa Fe Institute Artificial Stock Market Model (SFI-ASM) is one of the pioneering models and thus probably the most well-known and best studied. According to Waldrop (1992, p. 270), it emerged from a discussion between Ramon Marimon and Thomas Sargent on the one side and John Holland and Brian W. Arthur on the other side. Marimon and Sargent claimed that artificially intelligent agents in such a stock market simulation would quickly learn the neoclassical rational equilibrium solution.[1] Since Holland and Arthur could not agree with this hypothesis, they started in 1989 to develop the SFI-ASM which has been described by Palmer et al. (1994), Arthur et al. (1997), or LeBaron et al. (1999).[2]

Within the framework of the SFI-ASM, the Marimon-Sargent hypothesis comprises two parts. First, there should be no significant use of fundamental or technical trading information, and second, the price series should behave

---

[1] 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.

[2] There are, in fact, several 'generations' of the SFI-ASM on different programming platforms. An overview over the SFI market hi story 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 design of the original SFI-ASM model as reported in section 2 is based on the objective-C version 7.1.2, which also served as a blueprint for the author's own reprogrammed Java version using the RePast library. The source code is available on request.

"nicely", i.e., as predicted by the standard neoclassical solution. However, the model's main result 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 is characterized by more complicated price time series and by substantial levels of technical trading. Thus, for fast learning speeds both parts of the Marimon-Sargent hypothesis are rejected.

Arthur et al. (1997) asked themselves to

> *"what extent is the existence of the complex regime an artifact of design assumptions in our model? We find experimentally by varying both the model's parameters and the expectational-learning mechanism, that the complex regime and the qualitative phenomena associated with it are robust. These are not an artifact of some deficiency in the model."*
>
> Arthur et al. (1997, p. 35)

However, a closer investigation of the genetic algorithm (GA) that updates the trading rules of agents reveals that the SFI 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 this technical influence on the bit-level by an updated mutation operator, the Marimon-Sargent hypothesis is finally supported for most learning speeds. Agents now always discover the correct homogeneous rational expectation equilibrium (hree) of non-bit usage, no matter which GA-invocation interval is used. Furthermore, the simulated time series are generally closer to the hree-benchmark than in the original SFI-ASM. Only at much faster learning rates than in the original model can the claim of emergent complex price series behavior still be upheld.

In order to be self-contained, Section 2 will first introduce the basic structure of the original SFI-ASM with its main results. At the end of this section, the SFI mutation operator will be analyzed and its effects on the bit level will be illustrated. Section 3 develops an updated mutation operator. Finally, the results of the original and updated version of the SFI-ASM will be compared.

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 stationary AR(1)-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 expected utility function

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

with λ being the degree of risk-aversion and W_{t+1} being an agent's expected wealth level in the next period. 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

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

where $E_{i,t}[p_{t+1} + d_{t+1}]$ is the expectation in t about the next period's realization of the stock's price and dividend, and $\sigma_{t,p+d}^2$ is 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 balance the effective demands to the fixed supply of shares by setting 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 homogeneous 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 its forecast accuracy. 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 current price. When a particular predefined condition is met, the corresponding descriptor bit is set to 1, and 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, or holding # if the rule ignores that particular descriptor bit.[3] Rules with numerous #-signs are quite general, 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 have 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 to determine 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.

[3]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, the author will refer to them as bits.

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 the 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 when no rules match the market descriptor, 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 the time window that agents consider when estimating a rule's accuracy. As LeBaron et al. have pointed out, the value of θ is a crucial design question since it strongly affects the speed of accuracy adjustment and the resultant 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. At the other extreme, however, as θ goes to ∞, agents would take all past information into account, implicitly assuming they live in a static world. As in LeBaron et al., a value of 75 is chosen for θ.

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

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

with `specificity` being the number of conditions in a rule that are not ignored, `bitCost` being an associated cost for each bit set, and C being a positive constant to ensure positive fitness.[4] 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, LeBaron et al. claim that this should ensure 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 agents started with a rule set that contained only bad rules, in the absence of any other learning mechanism, they would not be able to find better ones. Thus, the GA provides a way to alter the rule sets by replacing the badly performing rules with new, possibly better ones.[5] By exploring the possible search space in a random, yet not directionless fashion, the GA creates the basis for further exploratory learning of the agents that occurs on a longer time scale than the accuracy estimation.

For each agent, the GA is, on average, invoked every K periods and replaces the 20 worst rules of the rule set. In doing so, the GA uses the genetic operators of *mutation* and *crossover*. Mutation is an important part of any evolutionary algorithm which helps maintain a diverse population and avoids premature convergence of the search algorithm. It could be interpreted as learning by experiment or by unintentional mistakes.[6] For predictor mutation Π, which is performed with a probability of 0.7 in the model, one parent is chosen by using tournament selection in which 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 bit mutation probability π 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

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

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

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

by adding random numbers to them.[7] 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 parents. Even though there are various 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 of 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.

As for the real valued parameters, LeBaron et al. 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 parent's values, with $1/\sigma^2_{j,p+d}$ as the weight for each parent. The weights are normalized to sum up to 1.

2.3 The Homogeneous Rational Expectations Regime

The normal model behavior for heterogeneous agents can be assessed by comparing it with the homogeneous rational expectation equilibrium. Since we know that all agents hold one unit of the risky stock in this benchmark scenario, the known structure of its dividend process allows us to solve for their hree-forecast parameters

$$a = \rho \tag{9}$$

and

$$b = (1-\rho)\left((1+f)\bar{d} + g\right), \tag{10}$$

with

$$f = \frac{\rho}{1 + r_f - \rho} \tag{11}$$

and

$$g = \frac{(1+f)(1-\rho)\bar{d} - \lambda\sigma^2_{p+d}}{r_f}. \tag{12}$$

[7] With probability 0.2 they are uniformly changed to a value within the permissible ranges of the parameter which is $[0.7, 1.2]$ for the a parameter and $[-10.0, 19.0]$ for the b parameter. With probability 0.2 the current parameter value is uniformly distributed within $\pm$ 5% of its current value, and for the remaining cases it is left unchanged.

The variance for the combined price plus dividend time series is

$$\sigma_{p+d}^2 = (1+f)^2 \sigma_\epsilon^2. \tag{13}$$

All information provided by the condition parts of the classifier system is unnecessary in the hree-case since agents only need to know the last period's price and dividend to derive their identical forecast for next period's price and dividend. Thus, the hree-solution should be characterized by complete negligence of technical and fundamental trading bits.

2.4 Experimental Results with the SFI-ASM

Depending on the GA-invocation interval, LeBaron et al. reported two different regimes. 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 1,000 periods. Bit usage remained low and the agent's forecast parameters converged to their hree-values, thus indicating that agents became more and more homogeneous.

The so-called rich psychological or complex regime arose when agents had a fast exploration rate, i.e., when the GA was often invoked, on average every 250 periods. Here, the continuously co-evolving agents remained heterogeneous with respect to their bit usage and forecast parameters. In fact, the emergence 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. Hence, the SFI-model also supports the "interacting agent hypothesis" as proposed by Lux (1998) and Lux and Marchesi (1999).

2.5 The Problem: A Faulty Mutation Operator

However, any interpretation of the simulation results that is linked to rule specificity and emergent technical trading has to be treated with caution in view of the mutation operator that has been used in the SFI-ASM. Unlike crossover, this mutation operator is not neutral to the initial level of bits set and usually introduces an upward bias in the resulting bit level.

In order to demonstrate this bit-increasing effect, we must examine the bit transition probabilities given in LeBaron et al. (1999, p. 1498). Once 0- or 1-bits are chosen for mutation with probability π, they are never left unchanged and are converted into the don't care sign # with a probability of two thirds. After mutation, an initial don't care bit will be either $1, 0$, or # with an equal probability of one third. LeBaron et al. assert that these transition probabilities would, on average, maintain the specificity, i.e., the fraction of #'s in a rule.

However, by applying a Markov chain analysis, we find that in the long run the fraction of non-# bits converges to one half. The transition probabilities given above can be expressed by the following transition matrix:

$$P = \begin{pmatrix} 0 & 1/3 & 2/3 \\ 1/3 & 0 & 2/3 \\ 1/3 & 1/3 & 1/3 \end{pmatrix}. \tag{14}$$

If we denote the vector of probabilities of the three possible states in period t as $\mathbf{p}^t = \{p_0^t, p_1^t, p_\#^t\}$, in equilibrium $\mathbf{p}^t = \mathbf{p}^t P = \mathbf{p}^{t+1}$ must hold. By repeatedly invoking the mutation operator, the vector of probabilities will converge to its equilibrium distribution of $\mathbf{p}^* = \left\{\frac{1}{4}, \frac{1}{4}, \frac{1}{2}\right\}$, i.e., on average, a quarter of all bits will be zero, another quarter will be one, and the remaining fraction of one half will be the don't care sign #.

Because the model usually functions well below the bit-level of one half, the mutation operator introduces an upward tendency in the bit distribution.[8] This is illustrated in figure (1) by varying the probability Π with

[8] The fact that the theoretical equilibrium level of 0.5 is usually far from being attained is due to a variety of other model parameters. First of all, even for a mutation probability of $\Pi = 1$, i.e., when there is no crossover in the model, every bit in the parent string will be changed with only a probability of $\pi = 0.03$. Secondly, the `bitCost` parameter penalizes every non-# bit, thus, the GA preferably selects rules with below average specificity for

which predictors are mutated in the model.[9] Note that throughout this paper, all simulation parameters have been set at the same values as those in LeBaron et al. (1999, p. 1492).

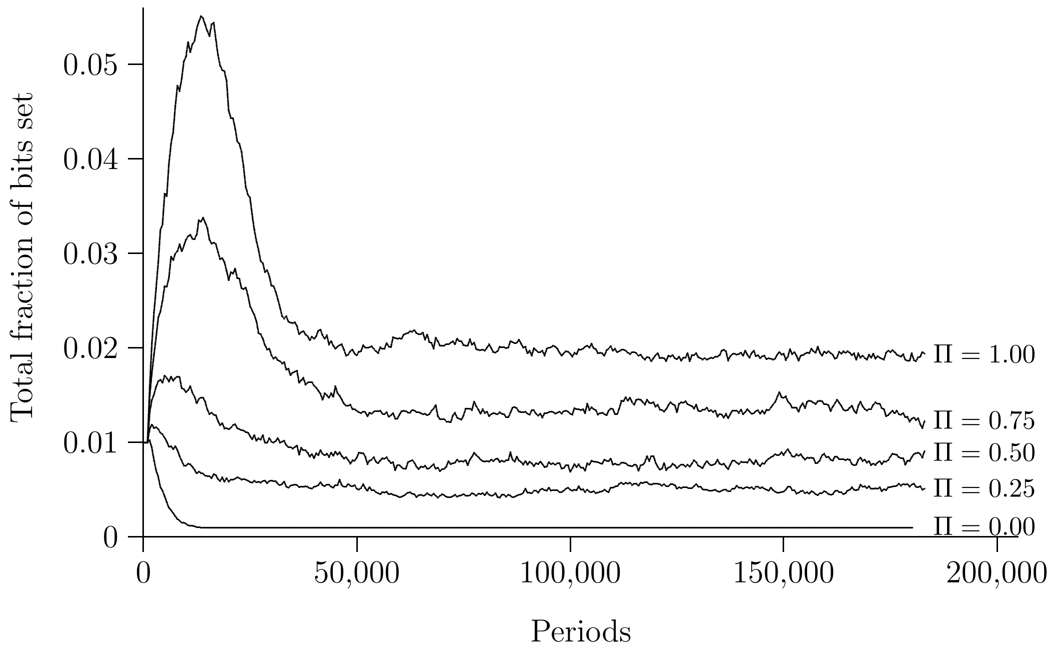

Fig. 1: Total fraction of bits set as a function of mutation probability Π in the original SFI-ASM. Data were obtained from a cross section of 5 separate runs at different random seeds.

Consequently, when increasing the learning speed, the mutation operator is more often invoked per time period and its upward bias results in a higher equilibrium bit level. The higher number of trading bits for faster GA-invocation intervals led the researchers from the Santa Fe Institute to

mutation and crossover. Finally, by looking at equation (5), we realize that the GA might produce illogical rules. 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 a probability of `genFrac`.

[9] A similar graph can be obtained by altering the bit mutation probability π.

infer that there is emergent technical trading in their artificial stock market. Because of the cost they have attached to every non-# bit, they conjectured that emerging trading bits must have, on average, some fitness-based advantages by producing more accurate forecasts.

However, there are several indications that this is a premature interpretation. First, their logic implies that the SFI-model could be forced into a non-bit (or at least a low-bit) usage solution if bit costs are sufficiently high. This, however, is never the case. Even for very high bit costs, only a minimal downward shift is noticeable in the equilibrium bit distribution for all mutation probabilities. Secondly, given the steady injection of new bits through the mutation operator, it is not surprising that one cannot find some kind of threshold level for the learning speed at which the model "jumps" between two distinct regimes. Instead, equidistant increases in the GA-invocation interval lead to similar increases in the equilibrium bit level. Furthermore, no significant differences in the level of fundamental and technical trading bits can be detected for various learning speeds or mutation rates. Last, but not least, one cannot assert that agents actually use technical trading bits simply because they exist. In fact, when counting an agent's rule usage, one realizes that general rules are much more often selected for use than more specific ones.

3 A Corrected Version of the SFI-ASM

3.1 Suggested Correction

The theoretical and experimental analysis above suggests that the emergence of technical trading bits may be an artifact caused by a misspecified mutation operator. Furthermore, the emergence of trading bits is not necessarily tantamount to emergent technical trading which requires that agents actually act upon these bits. In order to derive valid conclusions about the bit usage in the model, one should take care in designing bit-neutral operators and procedures. Thus, a strongly desired property of an alternative mutation operator would be that it leaves, on average, the fraction of set bits unaltered. While the bit-decreasing effect of the bit cost parameter is desirable as it is a fitness-based influence, the bit-increasing effect of the SFI mutation operator is undesirable since it is completely technical and economically not interpretable.

The suggested alternative bit-neutral mutation operator works with dynamically adjusted bit transition probabilities. In order to infer whether technical and fundamental bit usage differs in the stock market, this mutation operator works separately for fundamental and technical trading bits. Therefore, it is necessary to distinguish between the initial fraction of fundamental bits set $F_{fund.}$, and the initial fraction of technical bits set $F_{techn.}$. The transition matrix for the fundamental bits is then given by

$$P_{fund.} = \begin{pmatrix} 0 & F_{fund.} & 1 - F_{fund.} \\ F_{fund.} & 0 & 1 - F_{fund.} \\ \frac{1}{2}F_{fund.} & \frac{1}{2}F_{fund.} & 1 - F_{fund.} \end{pmatrix}. \tag{15}$$

and, similarly, for the technical bits by

$$P_{techn.} = \begin{pmatrix} 0 & F_{techn.} & 1 - F_{techn.} \\ F_{techn.} & 0 & 1 - F_{techn.} \\ \frac{1}{2}F_{techn.} & \frac{1}{2}F_{techn.} & 1 - F_{techn.} \end{pmatrix}, \tag{16}$$

It is easy to verify that these transition matrices ensure that $F^t_{techn.} = F^{t+1}_{techn.}$ and $F^t_{fund.} = F^{t+1}_{fund.}$, i.e., the fractions of set bits remain, on average, unaltered.

Having eliminated the upward bias of the SFI mutation operator, surviving trading bits should only emerge through competition and fitness considerations, implying that they indeed contain useful information.

3.2 Experimental Results

3.2.1 Agents Forecast Properties

An analysis of the real valued forecasting parameters as well as the mean variance of all rules does not exhibit any statistically significant differences from the original SFI model as reported by LeBaron et al. (1999, p. 1508–1509). This behavior was to be expected since the change in the mutation operator only affects the condition parts and not the forecast parameters. Thus, the differences for these parameters in the slow and fast learning regime must be caused by reasons other than a more or less extensive use of condition bits.

However, the bit usage between the original SFI-ASM and the updated model turns out to be strikingly different. Contrary to the original model, the updated model does not exhibit technical trading for all GA-intervals. In particular, one can see from figure (2) that most agents completely abandon technical and fundamental bit usage in the long run and do not check any conditions at all.[10]

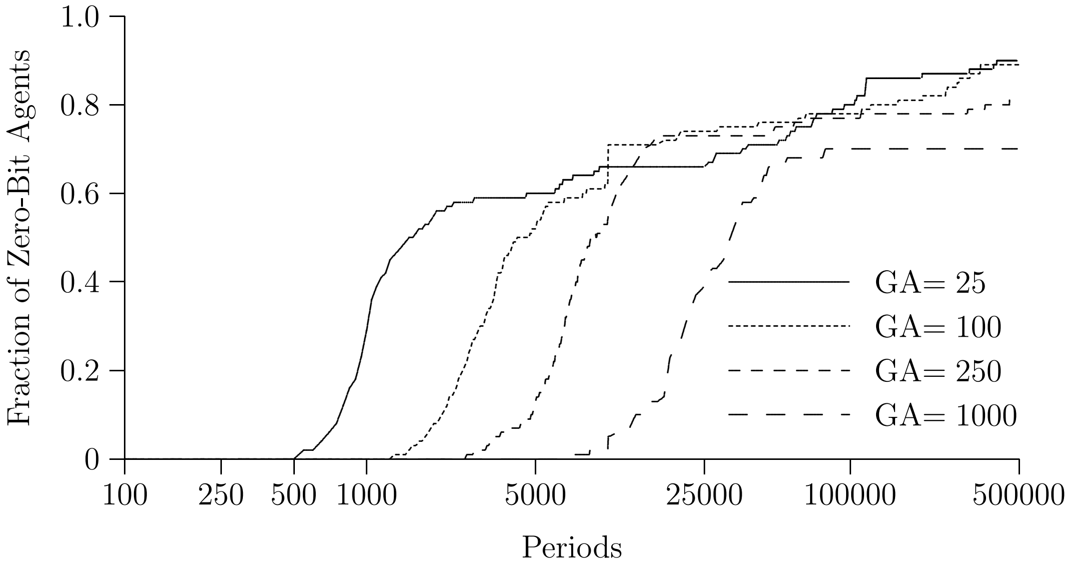

Fig. 2: Fraction of NESFI-Agents who discovered the correct hree non-bit usage solution (Zero-Bit Agents), recorded for different GA invocation intervals and averaged over 10 simulation runs.

Hence, replacing the original mutation operator with an updated operator that has the desirable property of being bit-neutral finally supports the first

[10]Some long run testing yielded that all agents will discover the correct non-bit usage solution. However, a few agents exhibit difficulties in doing so by being temporarily locked in a suboptimal solution, i.e., one or two specific trading bits in all their rules are set to either zero or one. Uniform crossover will always replicate this. Only mutation and generalization are able to change several of these bits in the pool of trading rules such that they can finally supersede the non-# bits. It is obvious that this may take a long time to happen.

part of 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. Even though this result is caused by just one relatively small change in the design of the artificial stock market, it is so radically different from the original model that I will henceforth refer to the corrected version as the NESFI-ASM (Norman Ehrentreich's SFI-ASM). Various tests have been performed with the NESFI-ASM so that the corrected results can be considered reliable.

First, in order to check the proper working of the classifier system, 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 the classifier mode was better for most GA intervals. Agents started to use some fundamental as well as technical bits while neglecting others, and were thus able to predict prices 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.

Second, one could argue that bit cost had been too high such that small efficiency gains from using condition bits had been overcompensated by their associated cost. However, for no bit cost at all, one notices that the fraction of fundamental or technical bits will either reach zero for some agents, or one for other agents. The overall fraction of used bits in the economy then stays constant. In order to explain this behavior, we have to realize that there are two corner solutions for which an initial bit distribution is exactly replicated. Because of the transition matrices (16) and (15), the bit distributions for the technical or fundamental bits get trapped once they reach either reached zero or one. The simultaneous existence of both corner solutions in the economy suggests that the GA has neither a built-in attractor towards which the equilibrium bit distribution is torn, nor that there are any fitness related gains or losses due to bit usage. In this case the bit usage of an agent follows a random walk, and, sooner or later, the fundamental or technical bit distributions for all agents will reach one of the two corner solutions.

Finally, a comparison of wealth levels of SFI-agents and NESFI-agents shows that, on average, both types of agents accumulate the same levels of wealth. Even in later periods of the market, i.e., when most NESFI-agents have abandoned bit usage, none of the two groups does better than the other. This is another indication that the classifier system in the artificial stock market does not provide any advantage or useful information that agents could exploit. Wilpert (2003, p. 128), too, concludes that giving up the condition parts of the trading rules does not have drastic effects on the model behavior. Thus, he also questions the usual interpretation of bit usage and emphasizes that its importance should not be overestimated.

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, it would also satisfy the second part of the Marimon-Sargent hypothesis by exhibiting "nice" price series properties. One should keep in mind that the proposed change to the GA only affects 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, p. 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. Therefore, they are regressed on a lag and a constant

$$p_{t+1} + d_{t+1} = a(p_t + d_t) + b + \epsilon_t, \tag{17}$$

and the estimated residual time series $\hat{\epsilon}_t$ is analyzed whether it satisfies being i.i.d. and N(0,4) distributed. The results are summarized in table (1).

First of all, one notices that the NESFI-ASM produces time series that are usually closer to the hree-benchmark than those of the SFI-ASM. The

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 1: 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.

standard deviations in the residuals are generally 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 more complex regime for faster learning rates than the original SFI-model.[11] The autocorrelation in the residuals, as shown in the

[11]While LeBaron et al. (1999) have reported the results only for the GA-intervals of 1,000 and 250, two additional learning speeds are included in table (1). The statistical tests were performed for even more GA-intervals, in particular, for 100, 50 ,25, 20, 10, 5, 2, and 1.

third row, demonstrates that there is little linear structure remaining except for the extreme case of updating the rule set in every period. As LeBaron et al. indicate, any artificial stock market should exhibit negligible autocorrelations since they are very low for real markets. The large autocorrelation coefficient for GA=1 indicates that the economic structure of the model might break down at this speed, i.e., equation (17) becomes misspecified.[12]

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 every period can extreme ARCH-behavior for all test runs be observed.[13] In row five, the first order autocorrelation of the squared residuals is another test for volatility persistence. Again, it increases for faster learning speeds 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.[14] 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 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 price series behavior can still be confirmed after the proposed change. However, it

---

[12]The autocorrelation coefficient for a GA-interval of 2 with a value of 0.06 (standard deviation 0.0035) is considerably lower than for an invocation interval of one. This supports the hypothesis that there is a structural break in the model when using the fastest possible learning speed.

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

[14]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.

is also apparent that for identical GA-invocation intervals, the NESFI-results are generally closer to the hree-benchmark.

An explanation for more efficient price series behavior in the NESFI-ASM 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 diverge from the hree-values, less efficient time series were likely to occur.[15] In other words, the larger the pool of activated trading rules to choose from, the larger the probability of choosing a good one.

Compatible with these results are the findings by Wilpert (2003, p. 128). He reports less kurtosis in the residuals and less trading volume when agents have no access to their classifier system in the first place. In the NESFI-ASM agents endogenously arrived at neglecting the classifier system and converged to the hree-solution.

4 Summary and Conclusion

In this paper it is shown that the feature of emerging technical trading bits in the Santa Fe Institute artificial stock market model was mainly caused by a faulty mutation operator which introduced an upward bias in the level of set trading bits. For faster GA-invocation intervals, the mutation operator was invoked more often, thereby increasing the equilibrium level of trading bits. Thus, emergent technical trading bits are an artifact of model design rather than a surprising result of interacting heterogeneous agents in the model.

When accounting for this problem by using a bit neutral mutation operator, agents completely abandon bit usage in the long run for all learning speeds. Thus, the Marimon-Sargent hypothesis stating that adaptive classifier agents in an artificial stock market will converge to the hree-solution

[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.

is supported with respect to bit usage. Agents realize that any temporary patterns they detect in the time series are random and not worth acting upon in the long run. Any additional information provided by the condition parts of the classifier system is neglected by the agents.

The situation is less obvious when analyzing the simulated price series of the corrected NESFI-ASM. At first sight, the second part of the Marimon-Sargent hypothesis also seems to be supported, since for identical learning rates, its price series exhibit fewer deviations from the hree-benchmark than those of the original SFI-ASM. Yet, when further increasing the learning speed, the model still shifts into a more complex regime with fat tails for the return distribution and higher trading volume. Hence, one could argue that the creators of the SFI-ASM were almost right, but for the wrong reason. Learning speeds much faster than initially thought can indeed trigger a more complex regime, but the latter must be caused by other reasons than an increase in technical trading bits.

Finally, this paper may also serve as an illustration of a more general problem in the field of agent-based simulation. Agent-based modelers usually face many design options. While it is possible to have several correct ways to tackle a particular problem, there are countless wrong or inappropriate approaches. This paper illustrates that special care is needed for even the most unsuspecting details of their model implementation. If a cause for a surprising model behavior cannot easily be found, agent-based modelers might be tempted to interpret this as emergent behavior while it may indeed be a result of an ill-designed model part. In order to allow for other researchers to replicate their research and to identify possible problems, it should become commonplace in the agent-based community to carefully document all design decisions as well as to provide their source code. Only because of this good practice, the author was able to discover the hidden problem in the original SFI-ASM.

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: W.B. Arthur, S. Durlauf, and D. Lane, (Eds.), The Economy as an evolving complex system II, Addison-Wesley, Reading, MA., pp. 15–44.

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

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

Dawid, Herbert, 1999, Adaptive learning by genetic algorithms: analytical results and applications to economic models, 2nd., rev. and enlarged edition, (Springer, Berlin et al.).

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, 425-441.

Goldberg, David E., 1989, Genetic algorithms in search, optimization, and machine learning, (Addison-Wesley, Reading, MA.).

Holland, John H., 1975, Adaptation in natural and artificial systems. (University of Michigan Press, Ann Arbor).

Johnson, P.E., 2002. Agent-Based modeling: what I learned from the artificial stock market. Social Science Computer Review 20, 174–186.

Kiyotaki, N., Wright, R., 1989. On money as a medium of exchange. Journal of Political Economy 97, 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, 1487–1516.

LeBaron, B., 2001. Empirical regularities from interacting long- and short-memory investors in an agent-based stock market. IEEE Transactions on Evolutionary Computation 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, 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, 329-373.

Mitchell, Melanie, 1996, An introduction to genetic algorithms, (MIT Press, Cambridge, MA.).

Palmer, R.G., Arthur, W.B., Holland, J., LeBaron, B., Tayler, P., 1994. Artificial economic life: a simple model of a stock market. Physica D75, 264–274.

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

Waldrop, M. Mitchell, 1992, Complexity: the emerging science at the edge of order and chaos, (Touchstone, New York et al.).

Wilpert, Markus, 2004. Künstliche Aktienmarktmodelle auf Basis von Classifier-Systemen, (Knapp, Frankfurt/M.).