CHALMERS

Internal Evolution for Agent Cognition

Agent-Based Modelling of an Artificial Stock Market

Master of Science Thesis in Complex Adaptive Systems

MORTEZA HASSANZADEH

Department of Energy and Environment
Division of Physical Resource Theory
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2011
Report No. FRT 2011:01

REPORT NO. FRT 2011:01

Internal Evolution for Agent Cognition

Agent-Based Modelling of an Artificial Stock Market

MORTEZA HASSANZADEH

Department of Energy and Environment

CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2011

Internal Evolution for Agent Cognition

Agent-Based Modelling of an Artificial Stock Market

MORTEZA HASSANZADEH

Technical report No. FRT 2011:01

Department of Energy and Environment
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Internal Evolution for Agent Cognition
Agent-Based Modeling of an Artificial Stock Market

MORTEZA HASSANZADEH
Department of Energy and Environment
Chalmers University of Technology

ABSTRACT

Agent-Based Modeling (ABM) is a powerful simulation technique with applications in several fields, in particular social sciences. Artificial Stock Market (ASM), introduced by a group of researchers at Santa Fe Institute (SFI) in the mid 90's, is one of the pioneering works in which the application of agent-based modeling is examined being used to model a stock market and to study economic behavior. A number of heterogeneous agents form a market in which they buy and sell shares of a single introduced asset and they make their decision upon their expectations of the market which is determined from their aggregate expectations, while they improve their predictions based on the response they get from the market. The computer simulation of the model gives a market dynamics which has similarities with real market data. The internal evolution of agents has a direct effect on the market dynamics, and the learning speed of agents controls the overall qualitative characteristics of the market. With emphasis on the role of evolution of agents in this artificial stock market, an implementation of the model has been done and a few issues have been studied.

Keywords: Complex Adaptive Systems, Agent-Based Modeling, Artificial Stock Market, Genetic Algorithms, Computational Economics.

ACKNOWLEDGEMENTS

Over the course of this thesis I had the opportunity to learn a lot from a group of nice people who made this work possible and it is a pleasure for me to thank them all. I am grateful to my advisor Kristian Lindgren, who suggested the subject for this work in first place and openly spent his time helping me along the whole way. I am thankful to Claes Andersson for the interesting discussions in the initial stages of this work. I would like to show my gratitude to Blake LeBaron who openly answered my questions about the model and kindly guided me. I am indebted to my friends for all their help. Many thanks to Mohammad Givehchi and Mohsen Tabareh who helped me in computer programming. I am also grateful to Reza Solgi who did me a great help in understanding analytical part of the work. I would also like to thank all people in Complex Adaptive Systems group at Chalmers University of Technology who let me learn many new and interesting issues during my master studies. Last but not least, I owe my deepest gratitude to my family for their love and unconditional support during different stages of my life including this level of education. This work could not be completed without their kind support and encouragement.

Gothenburg, Sweden
March 2011 Morteza Hassanzadeh

TABLE OF CONTENTS

1

Introduction

Complex adaptive systems, agent-based modeling, artificial societies, emergence, computational economics, evolutionary algorithms, and bounded rationality are topics that all are connected to an artificial stock market (ASM) model. Such a model is a good example of agent-based modeling in which a group of agents interact in a market to buy and sell shares and as a result, one can see how these rather simple agents can emulate a market which has some properties of a real market. A reimplementation of a model, in addition to the certain benefits that it can contribute to the model itself, can make one learn more about all topics involved in it. This work is no exception to this idea, especially regarding agent-based modeling which was the very first aim of this work.

In this chapter, complex systems and agent-based modeling are shortly explained mostly as to their terminological aspects. Other terms and expressions will be explained later when they are used. Then the literature corresponding to the model is reviewed followed by the organization of the whole work.

1.1. Complex Systems

A *complex system* is a system of many entities which act independently upon their own simple rules and interact with each other and with the environment without any central control in the system, though the collective behavior of the system is not only nontrivial and hardly predictable, but also exhibits one or more properties like emergence, self-organization, non-linearity, and chaos. Although there are several definitions by different scientists for the term "complex systems" and they seem to deviate especially in boundaries which are blurred enough to add complexity to the definition of complex systems itself, the general concept is the same. Ant colonies, nervous system and the brain, body immune system, economies, social structures, climate, modern energy systems, and a living cell are all examples of complex systems and have been studied.

Components of a complex system can learn from experience and evolve through a mechanism in order to improve their behavior and to adapt themselves to changes in a way they could be able to increase their success or the chance of survival. When this adaptation is significantly involved in the system and plays a large role, such systems often called *complex adaptive systems*. A stock market is a very good example of such systems; traders consider the market history and learn from their mistakes and successes, so they improve their predictions and make decision upon that. Most of the examples of complex systems mentioned before are also adaptive, like body immune system, nervous system and ant colonies.

Self-organization is one of the main properties of complex systems. The appearance of resulting global behavior from interacting individual entities as a coherent structure or pattern is called self-organization. Since there is neither an internal controller nor an external part that imposes a planned pattern, it becomes interesting. There are several examples of self-organization in different fields. A flock of birds is an example that you may encounter or already have seen in the nature. Each bird behaves according to three simple rules: separation, alignment, and cohesion. As a result, flocking behavior of birds gets form as a complex motion of the flock of birds which benefits them in several ways and would be very difficult to produce otherwise; imagine how hard it would be to plan such a motion individually for each part of such a system to get that result.

Emergence is another main property of complex systems which results from the interactions of the components. It also is defined by different scientist in different ways. Goldstein [1] defines emergent as "*the arising of novel and coherent structures, patterns and properties during the process of self-organization in complex systems*". More details of this definition are explained by certain common interrelated properties emergent phenomena share. They all are radically novel, coherent or correlated, global or at macro level, dynamical, and ostensive. There are also examples of emergence everywhere. Traffic jam is an example where cars, as parts of the system, behave on simple rules but the complex phenomenon of traffic jam sometimes emerges.

Considering these two, self-organization and emergence, Mitchell [2] brings an alternative definition for complex systems: "*a system that exhibits nontrivial emergent and self-organizing behaviors*".

1.2. Agent-Based Modeling

By the advent of fast computers, reductionism has been replacing with holistic approach to study systems. Although reducing a system to lower number of parameter has its own benefits and has been useful in the history of science and also is still used in many practical solutions, one can do a deeper study, thus more real and reliable, by considering details of the system as much as possible. Study of complex systems has been involved in this approach and agent-based modeling (ABM) is one of the best modeling techniques to do so. It relies on the power of computers and by taking details of the systems into account, provides the possibility to explore the dynamic of the systems which are impossible to explore by pure mathematical methods.

In ABM a systems is modeled according to a bottom-up approach; individual entities in the system are modeled as autonomous agents which can individually operate, thus providing a description of the system as it exist in reality. These agents can memories their experiences during the time evolution of the model. They can learn and adapt through mechanisms like classifier systems, artificial neural networks, and evolutionary algorithms. They make decision upon rules of their own and what they have learnt and then they behave individually as a result. A stock market is a good place for ABM to be applied. Traders are modeled as heterogeneous agents who learn from their experience and buy and sell shares and operate like real traders while a typical asset is introduced. The interaction of these agents makes a market interesting to be studied from different aspects.

ABM has important benefits. First of all, it can capture emergent phenomena. According to Bonabeau [3], the ability of ABM to capture emergent phenomena drives other benefits of it which are natural description of the system and flexibility. So, ABM is suitable to use when there is potential for emergent phenomena. On the other hand, the bottom up approach in ABM helps with understanding of what constitutes an explanation for such a phenomenon. This has been a great motivation for scientists to apply ABM to different fields such as social studies in order to approach social phenomena this way instead of using traditional modeling perspective. *Artificial societies* are result of applying ABM to social system and are used in experimental social studies, computational economics, urban modeling, and so on.

When the model is close to the natural description of the system, behavior of the individuals can clearly be defined. This also includes complexity in the behavior of individuals that might not be possible to consider in traditional modeling using differential equations which gives an exponential rise to complexity of the model and making it impossible to handle. Moreover, one can add stochasticity to the model wherever and in every level it is needed and this is not possible to do in an aggregate equation. Finally, crucial stage of validation and calibration is easier to do since an expert can understand the model easier and find correlations with reality.

Flexibility of the ABM is apparent. One can easily increase the number of agents, their behavioral characteristics, rules of their interactions, and levels of description. However, one should be careful when considering this flexibility since it may add more complexity than what is needed and may mislead the study without a precise and well planned investigation approach. This flexibility and natural description of the system can bring a load of parameters to the model which should be carefully considered in order not to lose tractability.

1.3. Literature Survey

It is about two decades now that computer simulations have made use of autonomous agents in economics and finance. During this period, many scientists have made attempts to answer questions in the rather new field of agent-based computational finance and economics. Artificial stock market modeling is the subject of study in this work but since the implementation of the model has been considered at first, a complete systematic literature review has not been aimed, though if not impossible, it might be very difficult to include all aspects of such a complex system. However, several articles have been considered in order to learn about the history of progressions in artificial stock market modeling.

The platforms and structures used in main researches in the field of agent-based computational finance are not common and a broad comparison is difficult as discusses LeBaron [4]. He summarizes six papers in details along with references to many other related works. Simple agent benchmarks, zero intelligent traders, foreign exchange markets and experiments, costly information and learning, and neural network based agents are all included in the studied literature in addition to the Santa Fe artificial stock market model which will be discussed more in details in this work and is one of the most adventuresome artificial market projects as LeBaron mentions.

In 1994, Palmer *et al.* [5] published an artificial stock market model. Using agent-based modeling techniques, artificial stock market was studied. The article explains how a number of simple agents can form an artificial market similar to the real market and notice the price dynamics that emerges as a result of the interaction of these agents. In this model, agents use classifier system to make their trading decision and they evolve using genetic algorithms.

Later, Arthur *et al.* [6], the same group of people, introduced SFI-ASM in which agents are able to forecast and make their decision to maximize their utility function upon their predictions instead of using classifier system of condition-action. Agents decide how much share they buy or sell based on their predictions. In each time step, the price is cleared through a mechanism based on the bid and offer in the market. The model is later explained more by LeBaron *et al.* [7]. It is shown that agents can adapt to the market of low learning rate and create a homogenous rational expectation regime while complex regime emerges when the parameter setting is changed and there is a higher learning rate. The model replicates time series properties of real market. Emergence of technical trading, temporary bubbles and crashes, are part of conclusions mentioned.

Palmer *et al.* [8] continue working on the model adding some modifications and changes to the parameter settings, *i.e.* providing the agents with more information. It is found that parameter settings in general and specifically learning speed play a key role in transition of the market from homogeneous rational expectation equilibrium to complex regime. In fact, the rational expectation equilibrium is a local attractor. They also suggest multiple stocks, impact of wealth, improved predictions, transition details, information control, and strategic behavior as subject of study and extension of the model for future plans at the time.

LeBaron [9] provides a guide for researchers interested in building their own artificial financial market. He outlines design issues involved on such a work, namely: agents, trading, securities, evolution, benchmarks or calibration, and time. Moving towards realism and validation, better understanding of generic properties of the system, make applications out of agent-based models, and entering to the other areas of economic and social science are possible directions for future of the researches on model that LeBaron concludes.

Time horizon is the main concern of LeBaron [10] while the structure of the model has many differences in comparison to original SFI-ASM. First of all, time horizon in which agents evolve, is studied as long- and short-horizon to see its effects on the results. The constant absolute risk aversion assumption is replaced with constant relative risk aversion so wealthier agents will have a greater impact on the market. A feedforward neural network with a single hidden unit is used instead of classifier-based trading rules. Finally, the mechanism for agents to acquire information is changed and a central pool of rules is used by agents instead of local rules of agents in original SFI-ASM.

Joshi *et. al.* [11, 12] use game theory to study SFI-ASM. The dominant strategy of technical trading arises due to a multi-person prisoner’s dilemma and results in a symmetric Nash equilibrium [11]. The learning speed is also studied from a game point of view and results show a unique strategic Nash equilibrium which is interpreted to be a sub-optimal equilibrium due to market inefficiency and lower earnings of traders [12].

Ehrentreich [15] suggest corrections to the SFI-ASM. He finds the mutation operator in the original model upwardly biased that suggests increased level of technical trading hence corrected version is unable to generate technical trading. However, two different regimes and other conclusions of the previous model are still partly present in results from his model. Later he focuses more on technical trading [14] and the issue of evolution in the model by providing tools to judge the relative importance of selection and genetic drift with arbitrary mutation operator. Ehrentreich later published a thorough exploration in the model and different issues regarding it as a book [15]. The book is not only an excellent reference to the both learning and empirical literature in finance, but also an important piece of work for understanding the dynamics of models with interacting learning agents as LeBaron mentions in the foreword of the book. In addition to what he discusses in [13,14], he also studies the dynamics of wealth and the complexity of that.

1.4. Organization of the Work

In this work, the attempt has been to implement the original SFI-ASM explained in [6, 7]. In chapter 2, the resulting implemented artificial stock market model and its structure will be explained in details. In chapter 3, the implementation of the model will be considered. It will also contain experiments done with the model and corresponding results and discussions. Finally, comes what is possible to conclude out of the whole work in chapter 4.

2

The Model

The basic structure of the model is more or less the same in various versions of the SFI-ASM. A number of agents interact with each other, buying and selling assets through a trading mechanism while an evolutionary algorithm helps agents to adapt themselves to the changes in the market and improve their behavior to maximize their utility function. In this chapter, the building blocks of the market will be introduced. In the organization of the building blocks, suggestions in [9] are considered. The market will be explained by its components. Starting from securities, agents will be explained later and the internal evolution of them will come in a separate section. Then trading mechanism will be discussed and the chapter will be finished with the time issues and the benchmarks. In each section, terms used will be explained to some extents subject to the constraint of scope of this work. Corresponding mathematical expressions will also be introduced. All of the mathematical expressions, if not stated, are borrowed from SFI-ASM literature [6, 7, 15].

2.1. Securities

The economic components of the market are set up to be as simple as possible. There are two assets present in the market. There is a risk free part of the market that pays cash money a constant interest rate of r from an infinite supply.

On the other hand, there is a single security in the market which pays agents a stochastic dividend in each period which is assumed to be a mean-reverting autoregressive -AR(1)- process as below

$$d_t = \bar{d} + \rho\left(d_{t-1} - \bar{d}\right) + \mu_t \tag{2.1}$$

in which d_t denotes the dividend at time t, $\bar{d}$ is the dividend mean, ρ is the speed of mean reversion, and μ_t is the stochastic shock which is chosen from a normal distribution with mean zero and variance σ_μ^2. This process is aimed for providing a large amount of persistence in the dividend process without getting close to nonstationary dividend processes [7]. The price p_t for the stock at each time step will be determined through trading mechanism which will be discussed later in this chapter.

2.2. Agents

Agents play role of traders who would be present in a real market. There are a number of agents (N), each has some properties and an internal evolution mechanism. With the aim of maximizing their utilities, they take actions at each trading opportunity, *i.e.* in each time step, and may either buy or sell shares. Thus they create a market which evolves along time and in which they interact. To make any decision, agents consider their preferences and act upon.

2.2.1. Preferences

Agents are constant absolute risk aversion (CARA) investors with the degree of risk aversion λ and a utility function of the form below[1]:

$$U(W_t) = -e^{-\lambda W_t} \tag{2.2}$$

W_t stands for the wealth of an agent at time t and they are homogenous with respect to their utility function that their degree of risk aversion is the same. They all are myopic in one period that they only consider their expectation of next period's price and dividend to maximize their utility function subject to their budget constraint:

$$W_{t+1} = x_t(p_{t+1} + d_{t+1}) + (1 + r)(W_t - p_t x_t) \tag{2.3}$$

In which x_t stands for the number of shares an agent owns at time t. It is known that under the normality assumption for the distribution of price and dividend, the amount of stock $\hat{x}_t$ that an agent desires to hold at time t is calculated as below:

$$\hat{x}_t = \frac{\hat{E}_t(p_{t+1}+d_{t+1})-(1+r)p_t}{\lambda\sigma^2_{t,p+d}} \tag{2.4}$$

In this equation, $\hat{E}_t(p_{t+1} + d_{t+1})$ indicates an agent's expectation at time step t about summation of price and dividend in the next period. $\sigma^2_{t,p+d}$ is the variance of time series of this combination of stock's price plus dividend, which is empirically observed while time evolves. This relation holds only under normality of the stock prices and this holds when there exists a rational expectation equilibrium. Otherwise, the distribution of the stock price is not clearly known and the connection to a CARA utility maximizer will be broken [7]. To form their expectations, agents use a forecasting mechanism based on the certain information they consider and keep updating.

2.2.2. Forecasting

Forecasting is possible by existence of a pool of rules that each agent has individually access to. Each rule consists of a set of basic properties: condition part, predictor, fitness, and forecast accuracy. The two parts of condition and predictor, together form the "condition-forecast" mechanism and are also called condition-forecast rules which is a modification of Holland's "condition-action" classifier system[2].

---

[1] Since the equations are given for each agent, all properties specific to agents would have an index i but is omitted for ease of notations unless it was necessary.

[2] An example of this classifier system can be found in [5].

The *condition part* of a rule is a vector which is checked against Boolean market state so has the same type and length as the market state. The market state is a vector of binary bits of market descriptors which reveals information from the current state and the history of the market. These bits may contain fundamental, technical, or any other information one would like to consider. The *fundamental* information considers dividend price ratio. For instance, the ratio of price multiplied by interest over dividend is compared to a number, say 0.75, and if it is greater the corresponding bit will take 1, otherwise zero. The *technical* information is based on the moving average type information. Whether the price is greater than the mean of price during last ten periods could be the information contained in one of the technical bits. Beside this, bits of the condition part are ternary and holding either 0, 1, or #. If a bit of rule aims to check the corresponding market descriptor and see if it matches the market descriptor or not, it will be set and will take either 1 or 0. If a bit contains #, it means that bit is unset and is ignoring the corresponding market descriptor, i.e. corresponding information, and it is called "*don't care*" bit in literature. Thus a condition part of a rule matches the market state if all 0 or 1 bits of it match the corresponding market descriptors. Table 2.1. shows typical examples of a few condition part with 12 bits checked against a typical market state.

Bits	1	2	3	4	5	6	7	8	9	10	11	12	matches ?
market state	0	0	1	1	1	1	0	1	1	0	1	0	-
rule 1	0	#	#	1	1	#	#	#	#	#	#	0	yes
rule 2	#	#	#	#	0	#	1	#	#	1	#	#	no
rule 3	0	#	0	#	#	#	#	1	#	#	1	#	no
rule 4	#	0	#	#	#	1	0	#	1	#	#	#	yes

Table 2.1. Checking condition part of 4 rules against market state.

Then comes the *predictor*. Predictor is a vector that accompanies the rule for prediction and forming expectation of the agent. In this model, a linear forecasting is chosen that using a vector of two real value elements, a and b, predicts the value for sum of price and dividend for the next period based on current period's values of them:

$$\hat{E}_t(p_{t+1} + d_{t+1}) = a(p_t + d_t) + b \tag{2.5}$$

Predictors are randomly initialized in a certain neighborhood and they will be changed and corrected while agents evolve.

After all agents are done with forming their expectations, price will be determined through trading mechanism and then agents can compare realized price and dividend with their expectations and measure how well their rules has made predictions. They calculate moving average of squared forecast error as below:

$$\nu_t^2 = \left(1 - \frac{1}{\tau}\right)\nu_{t-1}^2 + \frac{1}{\tau}\left[(p_t + d_t) - \hat{E}_t(p_t + d_t)\right] \tag{2.6}$$

And considering equation (2.5) it can be written as below:

$$\nu_t^2 = \left(1 - \frac{1}{\tau}\right)\nu_{t-1}^2 + \frac{1}{\tau}[(p_t + d_t) - a(p_{t-1} + d_{t-1}) + b] \tag{2.7}$$

This value is calculated for each rule of each agent separately. Parameter τ represents size of the time window that agents take into account to consider past information in calculating error variance. This value is used in 3 places. First, it is used as variance of rule's estimate which is used in the equation (2.4), thus

$$\sigma_{t,p+d}^2 = \nu_t^2 \quad (2.8)$$

Second, it is used to calculate fitness value for the rule:

$$f_t = K - \nu_t^2 - Cs \quad (2.9)$$

In this equation, K is an irrelevant constant, s stands for specificity which is number of bits set to either 0 or 1 that do not ignore corresponding market descriptor, and C is the levied cost per bit for specificity. This bit cost is purposed to get sure about the purposefulness of each used bit in terms of forecasting.

Finally, inverse of ν_t^2 indicates the forecast accuracy of the rule. Therefore, all elements of a rule are defined and they can be used as follows in the time section.

Each agent has a zero intelligence rule and keeps it along the time and if none of its rules got activated, it will enter the market and trade using this rule. The agent may update predictors of this rule but all bits of this rule will stay unset as are initialized so, at the beginning of the simulation. All other rules apart from this zero intelligence rule, will evolve and will be improved through an evolutionary mechanism.

2.3. Evolution

Agents learn and improve their behavior using information from the current state and the history of the market and through their own internal evolution. There is no direct communication between agents and their rules and they react to what other agents do only through the market and by means of information they get on price and dividend. Thus there is no imitative behavior present among agents. The evolutionary part of the internal process of agents uses a genetic algorithm that applies selection, crossover, and mutation to replace poorly performing rules with new better ones. To generate new offspring, with a crossover probability, which is a small number here like 0.1, the crossover operator is applied. Otherwise, the offspring will be generated by only mutation.

The *selection* is performed based on fitness of the rules. At each learning opportunity, a certain amount of badly performed rules regarding their fitness value will be selected and thrown away. Then to generate new offspring rules, parents are chosen by the selection which performs according to a simple tournament selection. Two rules are chosen completely randomly, and the better one will be selected with the probability equal to the tournament probability. This represents a tournament selection of tournament size 2. One can go further and increase the tournament size but it is not aimed in this model.

The *crossover* operator takes 2 parents each time and generates new offspring by doing a uniform crossover. As to condition part, this means that the crossover will perform bit by bit and for each new bit of offspring, it selects from either of the two parents with the same probability. This gives the condition part of the offspring which table 2.2. gives an example for.

bits	1	2	3	4	5	6	7	8	9	10	11	12
parent 1	#	#	1	#	#	0	0	#	#	0	#	1
parent 2	#	0	#	1	#	#	1	0	#	#	1	#
offspring	#	0	#	1	#	0	1	#	#	0	1	#

Table 2.2. An example of uniform crossover on condition part of the two parents.

As for prediction values there are 3 probable choices with the same probability:

(a, b) is selected from either of the two parents with the same probability.

a is taken from one parent and b from the other parent.

a and b are accuracy weighed average of parents.

The forecasting accuracy and error variance are simply set to the linear average of their parents.

The *mutation* operator works bit by bit on the rule and each bit evolves with a bit mutation probability which is a small number like 0.03. The mutation operator used in [6,7] is changed by Ehrentreich as introduced in [15]. But here, the original operator is used. Each bit is flapped according to bit transition probabilities given in table 2.3.

	0	1	#
0	0	1/3	2/3
1	1/3	0	2/3
#	1/3	1/3	1/3

Table 2.3. The transition probabilities for bits when they are mutated. The left most column corresponds to the bits of parent, and first row in top corresponds to the offspring.

The real valued prediction part, (a, b), may be replaced by either choosing a new random number in the range or taking from parent by adding a little perturbation. The probability for both cases are the same and is equal to 0.2. Otherwise, the value is simply taken from parent without any change. The variance of the offspring is set to the average of all rules of the agent. If this new value is less than the variance of the parent, less an absolute deviation, it will be set to the median of the all other rules.

2.4. Trading mechanism

Price dynamics is the main property of the market to be studied, and trading mechanism is going to directly influence it. Therefore it plays an important role in the model. There are three known trading mechanism used in the field as discusses LeBaron [9]. First method used in early models of stock market is a simple mechanism based on offer and demand. This mechanism acts once at each trading opportunity, i.e. time step, and increases the price in response to excess demand and vice versa as was used in [5]. Second, is the method used here in this work, in which a local equilibrium can be found using an iterative approach in the design of market and will be explained thoroughly later. Finally, comes the modeling of actual trading mechanism which needs learning and adaptation in the mechanism and beside its advantages, brings some complications into the model as well.

Trading happens according to the decisions agents make and the price is determined through an iterative procedure. A specialist takes offers and bids of all agents under current price which becomes first trial price. The sum of all agents desired shares should not exceed the sum of shares present in the market. In this model, one share is introduced for each agent at the beginning of the market, thus the following constraint helps the specialist to find the price:

$$\sum_{i=1}^{N} \hat{x}_t^i = N \tag{2.10}$$

If given price did not clear the market, the specialist calculates new price according to the following equation:

$$p_{trial}^{k+1} = p_{trial}^{k}(1 + \eta(B - O)) \quad (2.11)$$

In which B and O show the total bid and offer respectively, η is step length in iterative procedure, and k counts the iterations. If this new price does not clear the market, the iterations will continue until finding clearing price for the stock.

This mechanism resembles an auction for finding price. Specialist declares prices and agents resubmit bids and offers until the market is clear. Although it is not what happens in reality, the analogy between them can be explained as follows. The specialists in a real market have information from the past. They also use their experience, books, and other sources of knowledge and information. Therefore, they have a keen feel of demand function in their market [6].

2.5. Time

Evolutionary processes in the model give rise to different time issues like any other model in which agents and adaptation are present. One issue is the time horizons agents use to keep history and information from past periods of the market. This has been matter of question and different researchers has different beliefs on using short or long memory agents. In this work agents use a rather short memory and they are homogenous in this aspect. The market state, in fact, keeps a set of information which includes information from past 500 periods in the longest case, and reveals this for agents to use and agents are also designed to make use of this type of information only.

The second issue with time would be the speed of learning. How fast should agents adapt themselves with changes has a direct effect on the market dynamics. This will be discussed later in the experiments and one can see different speed of learning results in coherent differences in the market dynamics.

The third issue is synchronicity. In a real market, not all traders do trading in the same day or at the same time. They may do in different time occasions instead. But here in the model, agents are considered to take part in the trading all in every time period and synchronously. However, in each period, it is theoretically possible that some of the agents may appear not to trade according to their preferences and this may result in asynchronicity present in the real market. Moreover, asynchronicity is present in the evolutionary process and not all agents invoke genetic algorithm at the same time but according to a probability determined by learning speed.

A more general question about time would be the relation between time period in the model and the time in reality. Does a time period represent a day in reality? Then a 250000 periods would be about 685 years and 28 years if a period represents an hour. Such questions seem very important to be answered but they are contained in a bigger subject of validation which is still an open matter of discussion in both agent-based modeling in general and artificial stock market model specifically. Nobody has answered such questions thoroughly neither this work has aimed to.

It is also a good point to discuss the timing of the model. At the beginning of each time period, new dividend is introduced to the market and market state gets updated. Then agents start checking condition part of all of their rules against the market state. All rules matched the market state get activated. Each agent chooses the most accurate of its activated rules to form its expectation and enter the trading. They submit their bid or offer. The specialist. gathers all submissions and tries to find the best price that clears market according to the trading mechanism and agents desires may also get recalculated. When the new price is found trading stage is complete and agents update their properties, including wealth, cash money, and number of shares. They also calculate error of their activated rules and update their error variance and accuracy. Finally, they evolve subject to their learning speed and the new time period starts over.

2.6. Benchmarks

As already mentioned, validation and calibration contains an open set of questions in agent-based models of artificial stock market. The scope of this work is not defined to move towards answering questions of that type. But since a point in validation world is needed for any kind of modeling, the homogenous rational expectation equilibrium (HREE) is chosen here to be a criterion for the model and when the model gives results, they can be compared to. Thus a very first question in the model is whether agents can find HREE themselves and market converges to such equilibrium.

Since agents are homogenous with respect to their degree of absolute risk aversion, the homogenous rational expectation equilibrium (HREE) can be calculated considering linear price and dividend relation assumed in the forecast mechanism.

$$E_t^{HREE}(p_{t+1}+d_{t+1}) = a^{HREE}(p_t+d_t)+b^{HREE} \tag{2.12}$$

In which a^{HREE} and b^{HREE} are prediction real values corresponding HREE and can be analytically calculated. Considering the linear relation of price and dividend:

$$p_t^{HREE} = f d_t + g \tag{2.13}$$

parameters f and g shall be calculated. A homogenous equilibrium means all agents to hold an optimal amount of share, namely one share as their desired amount in equation (2.4). Thus one can write:

$$\hat{E}_t(p_{t+1}+d_{t+1}) - (1+r)p_t = \lambda\sigma_{t,p+d}^2 \tag{2.14}$$

The expectation of the sum of next period's price and dividend can be calculated as below:

$$\begin{aligned}
\hat{E}_t(p_{t+1}+d_{t+1}) = \hat{E}_t((1+f)d_{t+1}+g) &= g+\hat{E}_t\Big((1+f)\big(\bar{d}+\rho(d_t-\bar{d})+\mu_{t+1}\big)\Big) \\
&= g+\hat{E}_t\Big((1+f)(1-\rho)\bar{d}+\rho(1+f)d_t+(1+f)\mu_{t+1}\Big) \\
&= g+(1+f)(1-\rho)\bar{d}+\rho(1+f)\hat{E}_t(d_t)+(1+f)\hat{E}_t(\mu_{t+1})
\end{aligned}$$

Then considering that the expected value of μ for next time step is zero since $\mu_t \sim N\big(0,\sigma_\mu^2\big)$, one can end up with equation below for the expected value.

$$\hat{E}_t(p_{t+1}+d_{t+1}) = g+(1+f)(1-\rho)\bar{d}+\rho(1+f)d_t \tag{2.15}$$

And plugging this into the equation 2.14 and using 2.13 again, also noting that the result should hold for all times and the right side of the equation 2.13 is constant, finishes the solving of the equation for f and g as below.

$$f = \frac{\rho}{1+r-\rho} \tag{2.16}$$

$$g = \frac{(1+f)(1-\rho)\bar{d}-\lambda\sigma_{p+d}^2}{r} \tag{2.17}$$

And σ_{p+d}^2 can easily be calculated using equation 2.13 and taking the variance:

$$\mathrm{var}(p_t + d_t) = \mathrm{var}\big((1+f)d_t\big) \tag{2.18}$$

Thus

$$\sigma_{p+d}^2 = (1+f)^2 \mathrm{var}(d_t) \tag{2.19}$$

And from equation (2.1) and properties of a finite set of sentences from an AR(1) process, one can write:

$$\mathrm{var}(d_t) = \sigma_\mu^2 \tag{2.20}$$

Thus:

$$\sigma_{p+d}^2 = (1+f)^2 \sigma_\mu^2 \tag{2.21}$$

Therefore, the HREE forecast parameters which agents should seek in HREE regime are:

$$a^{HREE} = \rho \tag{2.21}$$

$$b^{HREE} = (1-\rho)\left((1+f)\bar{d} + g\right) \tag{2.22}$$

3

The Experiment

The vast number of parameters which usually exist in the agent-based models, contributes complexity to implementation of such models. This makes agent-based models very prone to generate delusive results. Moreover, the power of agent-based modeling relies mostly on computational approach thus making well implementation and management of parameters very crucial. In this chapter, starting from a word on implementation, the experiment design and the parameter setting will follow. Then results from the experiments will be discussed.

3.1. The implementation

The original model of SFI-ASM was implemented in Objective-C [6], and later it was converted to other programming languages like Java in [15]. Johnson [16] has developed a swarm version of the model and made it, in addition to other information and links to some other versions and updates, available on the internet[3]. MATLAB is also used by LeBaron [10] for implementation of the model.

In this work, there have been some exercises with MATLAB using *struct* to implement the structure of the model and properties of the agents. But there are good reasons to do such an implementation in an object-oriented programming language. It is easier to debug the program going through different classes and testing them independently in an object-oriented program rather than trying to check the flow of the data in procedural programming languages. Johnson concludes a few advantages of object-oriented programming in [16]. In object-oriented programming data is encapsulated into the classes and is retrieved when needed instead of leaving it about with the risk of being accidentally altered and working with classes is better than working with pointers. Therefore, C# is used as programming language and entities has been arranged in classes. Different objects are introduced as they exist in reality and in

[3] See: *http://artstkmkt.sourceforge.net*

the model. The isolation of different entities has been kept as will be seen later and this provides the opportunity to easily extend the model in terms of some parameters like number of stocks. Moreover, this makes it easier to manage variables and parameters in the model.

There are classes of *market*, *stock*, *agent*, *rule* and a class of evolutionary methods for agents to use. Agents have a number of instances of the *rule* class. They can execute methods like *trade* and by that, they submit their bids or offer to the mark, where the specialist stays. When the specialists declares the new price, agents' properties get updated and their rules get corrected and from time to time, upon the learning speed, they use methods in evolutionary class to improve their behavior.

3.2. Parameter settings

The main parameter setting in the work is borrowed from the original model discussed in [6], [7]. This not only makes the replication of the original model possible, but also provides the possibility of comparing the results. Since different entities are put into different classes in the implementation, parameter values are arranged and introduced here according to the same classification. This makes it easier to compare the report and the information in it with the implementation and also simplifies any new implementation.

The market class simply includes agents and assets. There is also a specialist who at each time period uses the data from the whole market and finds the market clearing price through a number of iterations. This number was put to 10 in the original model [6] but here since the speed of computers allow, it try much more time to find a market clearing price. However, if this did not happen, the best try will be declared as price for that period. Moreover, since a mathematically exact number for price in a computational method is out of reach, trials of the specialist is checked against a threshold (δ_x) for the difference between sum of agents' desired number of shares in equation 2.10 and the total number of shares in the market, N. The step size in price finding iterations is calculated after running the program for some times and measuring the number of times that market clearing price is found. Table 3.1 represents all initial values for parameter setting of the market.

Parameter Description	Notation	Value
Number of agents	N	25
Number of holdings (with stochastic paying)	-	1
Number of risk free bond	-	1
Interest rate for risk free bond	r	0.1
Number of iterations for finding market clearing price	-	200
Threshold for cleared market	δ_x	1

Table 3.1. Parameter values for the market in the simulation.

The market will not allow agents to go short for more than 5 shares in each time step. There is also a limit considered for the maximum price which is 200 and off course the price cannot be negative. These assumptions have appeared to have a small effect only at early stages of the market and they do not influence results.

Agent is another class and includes some other parameters. They are homogenous in terms of their risk aversion degree which is same for all. The number of rules for agents are also the same. All corresponding parameters can be found in table 3.2.

Parameter Description	Notation	Value
Number of rules	M	100
Risk aversion degree	γ	0.5
Initial cash	-	2000
Initial number of share	-	1

Table 3.2. Parameter values for agents in the simulation.

Rules are randomly initialized according to the mechanism described in previous chapter (see section 2.2.2). The corresponding values in their initialization and other parameters come in the table 3.3.

Parameter Description	Notation	Value
Number of condition bits	J	12
Probability of turning a bit on: equal to 1	-	0.05
Probability of turning a bit off: equal to 0	-	0.05
Real value predictor range	a	$[0.7, 1.2]$
Real value predictor range	b	$[-10, 19]$
Initial value of error variance	$\nu_{\circ}^2$	4
Time window size	τ	75
Cost per bit usage	C	0.005
Fitness constant	K	100

Table 3.3. Parameter values for rules in the simulation.

There is one holding instance of the stock class and it pays a stochastic dividend based on an AR(1) process and using parameter values as in table 3.4.

Parameter Description	Notation	Value
Number of holdings	$-$	1
Speed of mean reversion	ρ	0.95
Dividend average	$\bar{d}$	10
Variance of stochastic shock	σ_{μ}^2	0.0744

Table 3.4. Parameter values for stock in the simulation.

These parameters form the input to the model and then one can test to see if the model simulation confirms known basics and here, HREE can be interpreted as one of the outputs that can evaluate the simulation if appears in the results. Table 3.5 represents parameter values which are calculated analytically and form the expectations here from the simulation.

Calculation of the parameter	Notation	Value
$\rho/(1+r-\rho)$	f	6.3333
$((1+f)(1-\rho)\,\bar{d}-\lambda\sigma_{p+d}^2)\,/\,r$	g	16.6667
ρ	a^{HREE}	0.95
$(1-\rho)\left((1+f)\bar{d}+g\right)$	b^{HREE}	4.5

Table 3.5. Analytically calculated parameter values.

It would be beneficial to mention that the term σ_{p+d}^2 in calculation equation of g is calculated using equation 2.21 when f is calculated.

Another part of initial settings belongs to the definition of the market state. The market state can include as much information as one may wish to. Joshi *et. al.* [11, 12] use 64 bits including 3 dummy bits and Ehrentreich [15] includes 32 bits of both fundamental and technical bits, so 64 bits in total. Here, same as the original model, 12 bits of information are included in the market state. 6 fundamental bits and 4 technical bits have been used while last two bits are always set to one and zero regardless of what is happening in the market. Table 3.6 represents all information included in the market state of the model used for simulation in this work.

Condition	Bits
Price × interest / dividend > 1/4	1
Price × interest / dividend > 1/2	2
Price × interest / dividend > 3/4	3
Price × interest / dividend > 7/8	4
Price × interest / dividend >1	5
Price × interest / dividend > 9/8	6
Price > 5-period moving average	7
Price > 10-period moving average	8
Price > 100-period moving average	9
Price > 500-period moving average	10
1	11
0	12

Table 3.6. Condition bits of the market state.

Finally, different values in evolutionary process of the program, especially the learning speed, play an important role in the simulation. Table 3.7 shows different parameter setting in genetic algorithm used for evolution of agents.

Parameter Description	Notation	Value
Crossover probability	P_c	0.1
Mutation probability	P_m	0.9
Bit mutation probability	P_{bm}	0.03
Percentage of worse rules to be replaced	-	20

Table 3.7. Parameter values for evolutionary mechanism in the simulation.

3.3. Design of the experiment

Since the first aim of this work has been replicating the original model, the first experiment, same as in the original model, is to check if the model can reach the homogenous rational expectation regime as discussed in the benchmark section in 2.6. To do so, the model is first run for 250000 periods to let the agents and the market have enough time to evolve and then the price series is recorded for 10000 periods and is compared to the linear price dividend assumption in first place. Recalling predictors used for prediction one can write:

$$p_{t+1} + d_{t+1} = a + b(p_t + d_t) + \varepsilon_t \tag{3.1}$$

In this equation, ε_t represents residual value and will be used to analyze the structure. This experiment is done for two learning speeds; the genetic algorithm is invoked every 1000 periods for slow learning regime and every 250 period for the fast learning regime. This learning speed, as will be seen in the results, can highly affect the result and move a homogenous rational expectation equilibrium towards a complex regime. However, the effect of this parameter is not thoroughly studied here and the question of the optimal learning speed or the range of learning speed corresponding with transition from an equilibrium to a complex regime is left unanswered here but one can find more on this in [15].

3.4. Results

There are many different aspects that the model can be viewed from. Basic statistical analysis has been the first method to study results from early implementation and comparing them with real market data hence evaluating the model and implementation as a whole. Price dynamics has been the first outcome of the model to be studied but it can be followed by many other properties defined in the model. Trading volume, effect of wealth, level of bit usage, the evolutionary mechanism, and effect of learning speed are all examples of data that can be extracted through simulation for any directed analysis. Moreover, other questions can be asked and the model can be tailored to answer them, like game theoretic questions discussed in [11, 12].

Here, with the limited scope of this work, price dynamics is plotted out and will be discussed. The level of bit usage is another data extracted through simulation and will also be studied. Finally the predictors of the agents will be investigated.

3.4.1. Price dynamics

Resulting price dynamics for different scenarios are illustrated in figure 3.1. As can be seen, price, similar to the real market data, appears to be persistent in the results from simulation while following the persistency in dividend series. In comparison with the fast learning regime, slow learning agents seem to be much more successful to follow the REE benchmark. This benchmark is calculated and plotted based on the given input dividend series and the linear relation assumed between dividend and price in REE (see equation 2.13). When the genetic algorithm is invoked more frequently, while all other parameters are kept the same, the system exhibits different behavior which is apparent in terms of expected variations for the price series shown in the figure. Other properties of the fast learning regime deviates more from the benchmark. Basically, such differences has been the reason for naming the slow learning regime the “HREE” regime and calling the fast learning regime "complex" in the corresponding literature and by main authors.

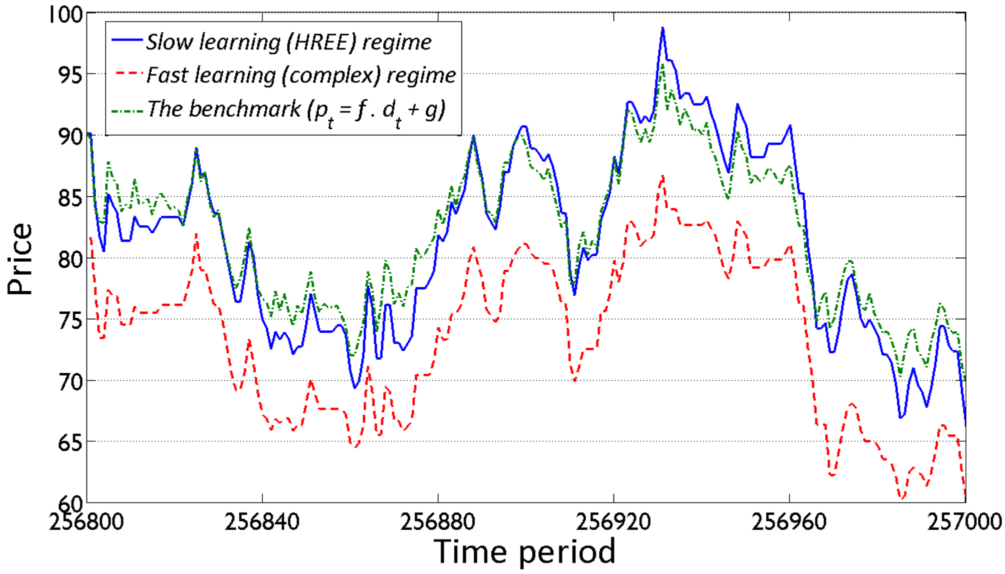

Figure 3.1. Price dynamics for different learning speeds. Faster learning speed results in complex regime in which the price dynamics is far from the homogenous rational expectation equilibrium.

When analyzing ε_t as already discussed, the resulting standard deviation is a bit different from what is mentioned in the literature regarding the original model. However, it confirms that in the fast learning regime, results deviate more from the benchmark. The standard deviation for slow learning regime is closer to the benchmark than the complex regime. It also shows less excess kurtosis than the complex regime. Table 3.8 compares results from the model with the original model [7].

Description	Slow learning		Fast learning	
	SFI	This work	SFI	This work
Standard Deviation	2.135	2.054	2.147	1.877
Excess Kurtosis	0.072	0.72	0.320	1.092

Table 3.8. Two statistical properties of the $\boldsymbol{\varepsilon_t}$ extracted from price series compared to the results from original SFI-ASM.

3.4.2. Bit usage

The fraction of bits set in the market by agents has been another output data of ASM studied here. Recalling the definition, a bit is set when it carries information and is set to either 0 or 1. In the original SFI-ASM, there was an emergence of technical trading in complex regime recognized by authors [6]. But later, Ehrentreich brought another implementation which did not confirm that but the other conclusion regarding bit usage in the original model: there are higher level of aggregate bit usage when the agents learn and evolve on a faster pace. However, results here do not confirm neither of them. Figure 3.2 shows how technical bits are going to settle down in both regimes and on a lower value for the faster one. The moving average of the total fraction of bits set are plotted for both regimes.

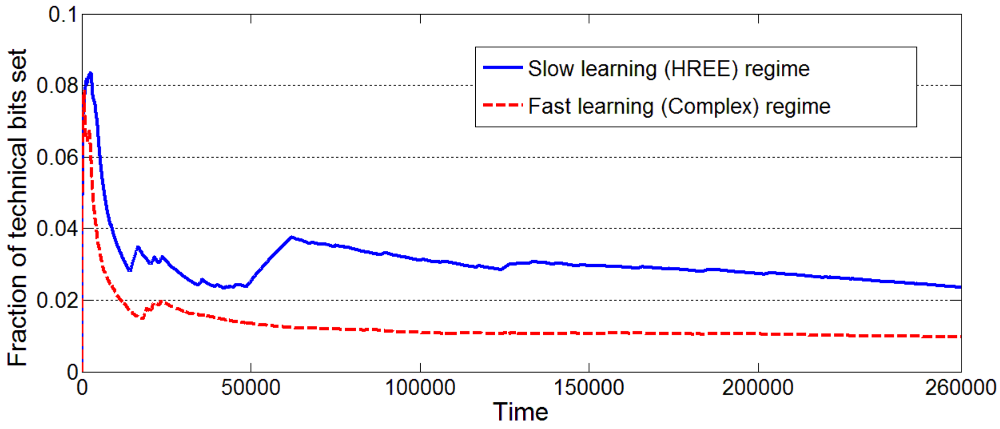

Figure 3.2. The moving average of total fraction of technical bits set.

Figure 3.2 shows that agents tend to use technical information less often. However, it does not mean that they do not use the information they have. They use fundamental information they have also the useless information of last two bits of condition part of their rules. These can be seen while having a look on illustrations of fraction of different kinds of information bits set in figure 3.3.

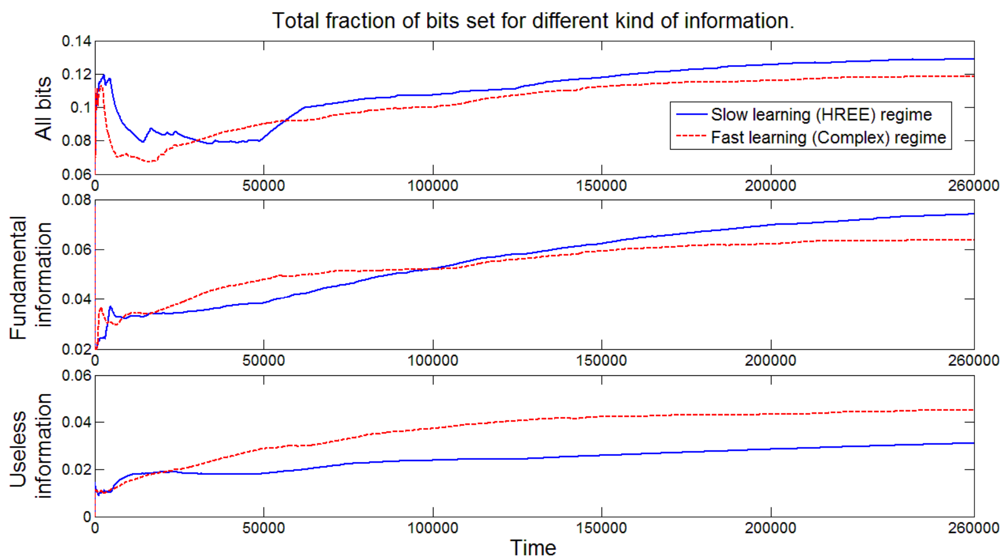

Figure 3.3. The moving average of total fraction of bits set for different kind of information.

3.4.3. Predictors

Prediction parameters, a and b, are randomized in ranges $[7, 1.2]$ and $[-10, 19]$ respectively. While the market evolves along time, the winning rules prediction parameters line up on a direction passing through the center of the region, which basically is the point where parameters corresponding to the benchmark exist. In the slow learning regime, they line up slowly but on an order that all points seem to belong to the same line. In the fast learning regime, they soon lose a bit of their order on the line and some of the predictors stay away from the line and also from the center of the region. Figure 3.4 shows snapshots of the distribution of the prediction parameters of all agents at different time steps for both regimes.

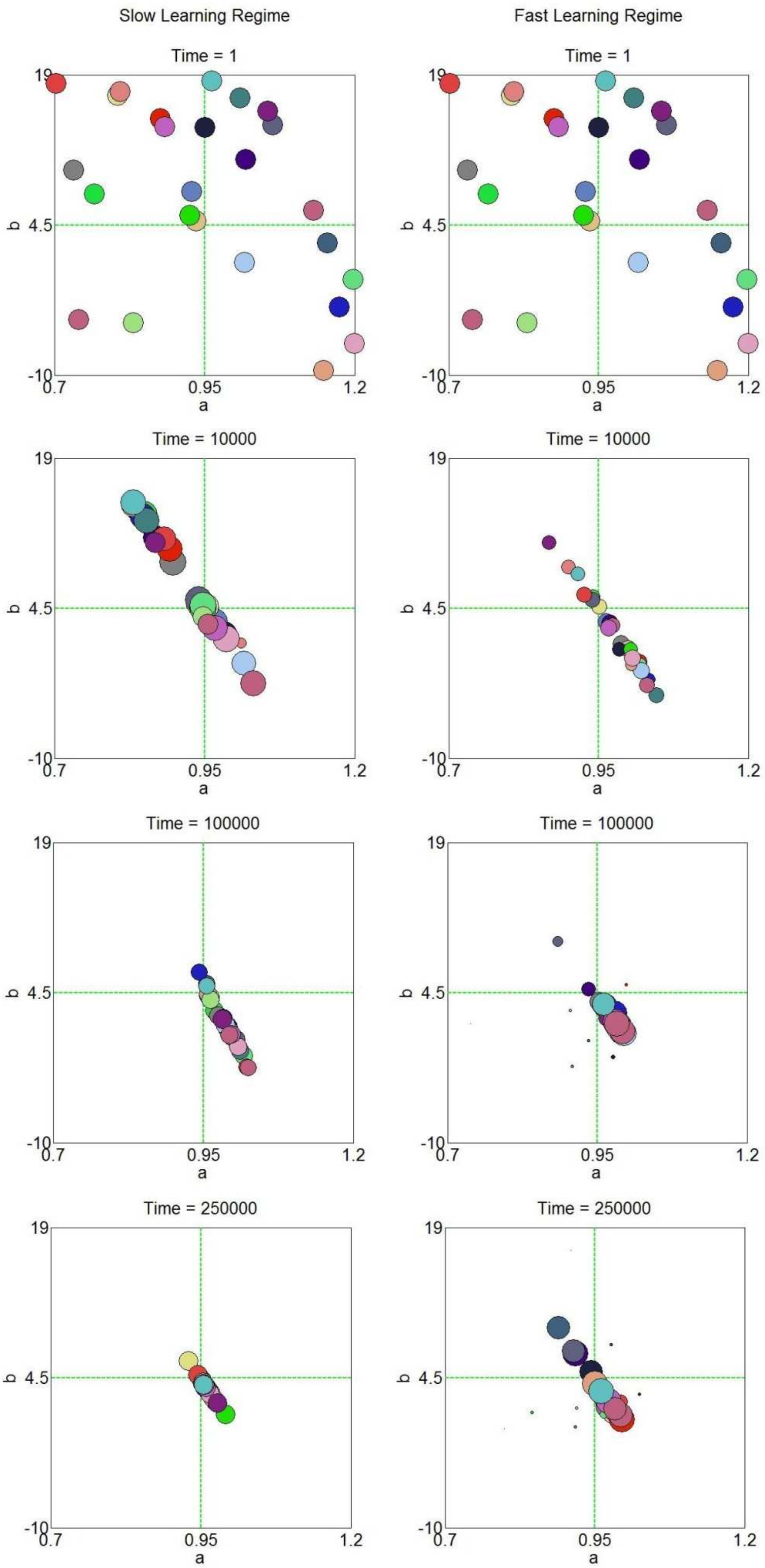

Figure 3.4. The evolution of winning predictors. The size of each circle represents the accuracy of the predictor.

3.5. Discussion

Stylized properties of the overall outcome of a system of simple individuals has been one of the interesting subjects to study and agent-based modeling of an artificial stock market is one of the best examples for that in the artificial world. The implementation shows how some simple agents without any communication with each other and relying only on their internal evolutionary mechanism and through interactions with the environment, can construct a market with stylized properties of a real market. The internal evolution of the agents hence becomes the mean for their cognition and given a sufficient amount of time, those simple agents can become members of a rationally performing market formed by themselves.

It is apparent from figure 3.1 that agents have constructed a market similar to the real market. They have been successful to follow the dividend variations and reach the benchmark when the learning speed is slow. The risk aversion considered for the agents seems to result in a lower price in fast learning regime. The standard deviation of the price in slow learning regime seems very close to what is expected, *i.e.* 2. There are also more excess kurtosis than in results from the original SFI-ASM. These differences would be due to some differences in parameters used in the implementation like δ_x in table 3.1. There is also a small difference in the variance of the random shock in the input dividend series.

Studying fraction of the bits set shows agents continuously use information they have while each agent always owns a zero intelligence rule; a rule without any bit set. This may infer that zero intelligence agents would not be as successful as current agents at least in terms of generating a rational market.

The order appeared in predictors' space in figure 3.4 seems interesting. Instead of gathering around the center of the region, they line up on a direction passing through the center. Increasing the learning speed, has not leaded agents to achieve the predictors of the benchmark. Instead, it has resulted in scattering of a few of predictors and this may explain the deviation the complex regime has from the benchmark. On the other hand, since there seems to be a linear relation between parameters a and b, it seems that the interacting mechanism of agents results in elimination of one of the prediction parameters, as it can be calculated upon the other one. This is more true for slow learning regime than for the complex regime where a few scattered points are present. Figure 3.4 shows snapshots of the distribution of the prediction parameters of the agents at different time steps for both regimes.

4

Conclusions and Future Work

SFI-ASM might be the most difficult artificial market model to implement. There are too many parameters and they all need to be treated carefully. Although the calibration of the model at its current level was previously done by main authors of the SFI-ASM, there still are some considerations even at this level. Price finding mechanism is one issue to be considered. Price is calculated through an iterative procedure using equation 2.11. It needs a thorough attention finding a suitable step length, η, in that equation. This in addition to other problems with the SFI-ASM is well discussed by LeBaron in [17]. Another issue with the mechanism which specialist uses to find the market clearing price, happens when setting the threshold for the equation. Solving such an equation computationally needs taking a threshold, δ_x, by which the market is supposed to be cleared. When setting this number equal to one, the simulation goes on well and one can get results as in this report. But, this threshold is equal to 4 percent of the total amount of stock and if one considers trading volume, it is more than trading volume in some periods or at least the error it contributes is considerable. Therefore it seems to be crucial to increase the resolution. However, doing so, results in unstable market with an oscillating price dynamics. This has been the case when $\delta_x = 0.01$. This, beside other consideration, may affect the results in one way or another. Therefore, one should consciously have them in mind and reconsider them before any generalization about results. However, there is no direct way to eliminate all errors or to instantly measure the dimensions and effect weight of each parameter. Thus further research and investigations are needed in this regard.

4.1. Conclusions

It can be concluded that agents in existing mechanism and settings, have succeed to reach homogenous REE themselves. In addition to that, when they learn faster, they can form a market much more similar to a real market. But how similar is the dividend series to empirical data and how much applicable the result would be in dealing with real world financial problems are questions of calibrating with real data that still remain unanswered for some reasons. Shortcomings of the model as counts LeBaron [17], economically uninteresting properties of the model, and more importantly the availability of other modern models have made details of this model less interesting.

Study of bit usage shows that agents use information they have access to. Thus it might be better to use more information and longer bit strings for condition part, as did Joshi *et. al.* and Ehrentreich in their works [11-15]. However, this needs some considerations and first of all, desired goals and specific questions that the model would be expecting to answer shall be defined first. Otherwise, the sensitivity of the model to parameters could make such a change less helpful. Ehrentreich has answered a question of the type in [15] after well defining the question.

The evolutionary part of the model remains the most interesting part and plays the role of a heart to body of the model. Although Ehrentreich [13, 15] suggests some corrections in the genetic algorithm mechanism and specifically the mutation operator, the existing mechanism has also been successful. The interpretation of the order appeared in the predictors arrangement needs more investigation to see if it is a design consequence or not.

Finally, to learn and deeply understand the agent-based modeling, SFI-ASM is a very good choice and provides one with the opportunities to learn a lot about details of planning the whole system, evolutionary mechanism, and programming issues. However, putting some additional goals into the context, by advancing the model for instance, would make the whole work more inspiring and would result in more promising conclusions.

4.2. Future works

It is necessary to conduct a complete set of statistical tests on cross section of a sufficient number of runs to have more robust results as previously done in corresponding literature. This can provide more reliable comparison of this implementation with original model. Then it would be more interesting to investigate the order appeared in predictors and to see if it can be of any specific interpretation. It is also worth to check trading volume again as well.

Further research possibilities of the model itself has already been well described by Palmer *et. al.* in [8]. They suggest using multiple stocks in the model, impact of wealth, improved predictions, transition details, information control, and strategic behavior as subjects of study and extension of the model for future plan. However, LeBaron, one of the main builders of the SFI-ASM has stopped working on this model after about a decade research on that [17] and has moved to other modern models. If not stopping the work on this model, it is necessary to find a good way to correct or improve inefficiencies of the model and to enrich financial aspects of it. In that case, it may not be called SFI-ASM anymore and this may already has happened as Ehrentreich mentions in his book [15]. But the main idea and structure of the model will keep its uniqueness.

It would also be interesting to use possibilities at this stage, and to be specific, the internet to put the model available online through applications so that human can enter the market as users in a game or so. This would provide the opportunity to conduct studies on a market that partially benefits real traders' behavior. A work of the kind is done by Gulyás *et. al.* and described in [18, 19].

References

[1] J. Goldstein. Emergence as a Construct History and Issues. *Emergence*, 1(1): 49-72, 1999.

[2] M. Mitchell. *Complexity A Guided Tour*. Oxford University Press, 2009.

[3] E. Bonabeau. Agent-Based Modeling: Methods and Techniques for Simulating Human Systems. *PNAS*, 99(3): 7280-7287, 2002.

[4] B. LeBaron. Agent-based computational finance: Suggested readings and early research. *Journal of Economic Dynamics and Control*, 24(5-7):679-702, 2000.

[5] R. G. Palmer, W. B. Arthur, B. LeBaron, and P . Tayler. Artificial economic life: a simple model of a stockmarket. *Physica D*, 75:264–274, 1994.

[6] W. B. Arthur, J. H. Holland, B. LeBaron, R. Palmer, and P. Tayler. Asset pricing under endogenous expectations in an artificial stock market. In W. B. Arthur, S. N. Durlauf, and D. A. Lane, editors, *The Economy as an Evolving Complex System II*, volume 27 of *SFI Studies in the Sciences of Complexity*, pages 15–44. Santa Fe Institute, Addison-Wesley, 1997.

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

[8] R. G. Palmer, W. B. Arthur, J. H. Holland, and B. LeBaron. An artificial stock market. *Artificial Life and Robotics*, 3(1):27–31,1999.

[9] B. LeBaron. A builder's guide to agent based financial markets. *Quantitative Finance*, 1(2):254–261, March 2001.

[10] B. LeBaron. Evolution and time horizons in an agent-based stockmarket. *Macroeconomic Dynamics*, 5(2):225–254, 2001.

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

[12] S. Joshi, J. Parker, and M. A. Bedau. Financial markets can be at sub-optimal equilibria. *Computational Economics*, 19(1):5–23, 2002.

[13] N. Ehrentreich. The Santa Fe Artificial Stock Market reexamined: Suggested corrections. Betriebswirtschaftliche Diskussionsbeiträge 45/02, Martin Luther University Halle-Wittenberg, 2002.

[14] N. Ehrentreich. Technical trading in the Santa Fe Institute Artificial Stock Market revisited. *Journal of Economic Behavior and Organization*, 61(4):599–616, 2006.

[15] N. Ehrentreich. *AGENT-BASED MODELING The Santa Fe Institute Artificial Stock Market Model Revisited.* In M. Beckmann, H.P. Künzi, editors, Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin, 2008.

[16] P. E. Johnson. Agent-basedmo deling: What I learnedfrom the artificial stock market. *Social Science Computer Review*, 20(2):174–186, 2002.

[17] B. LeBaron. Building the Santa Fe Artificial Stock Market. Working paper, Brandeis University, 2002.

[18] L. Gulyás, B. Adamcsek, and A. Kiss. An early agent-based stock market: Replication and participation. *In Proceedings 4th Soft Computing for Economics and Finance Meeting*, May 29, 2003,University Ca' Foscari, Venice, Italy, 2003.

[19] L. Gulyás, and B. Adamcsek. Charting the Market: Fundamental and Chartist Strategies in a Participatory Stock Market Experiment. *DEVELOPMENTS ON EXPERIMENTAL ECONOMICS*, Lecture Notes in Economics and Mathematical Systems, 590(V): 219-224, 2007.