An Heterogeneous, Endogenous and Co-evolutionary GP-based Financial Market

Serafin Martinez-Jaramillo, Edward P. K. Tsang,

Abstract— **Stock markets are very important in modern societies and their behaviour have serious implications in a wide spectrum of the world's population. Investors, governing bodies and the society as a whole could benefit from better understanding of the behavior of stock markets. The traditional approach to analyze such systems is the use of analytical models. However, the complexity of financial markets represents a big challenge to the analytical approach. Most analytical models make simplifying assumptions, such as perfect rationality and homogeneous investors, which threaten the validity of analytical results. This motivates alternative methods.**

In this work, we developed an artificial financial market and used it to study the behavior of stock markets. In this market, we model technical, fundamental and noise traders. The technical traders are sophisticated genetic programming based agents that co-evolve (by means of their fitness function) by predicting investment opportunities in the market using technical analysis as the main tool.

With this endogenous artificial market, we identified conditions under which the statistical properties of price series in the artificial market resembles those of the real financial markets. Additionally, we modeled the pressure to beat the market by a behavioral constraint imposed on the agents reflecting the Red Queen principle in evolution. We have demonstrated how evolutionary computation could play a key role in studying stock markets.

Index Terms— **Finance, Genetic Programming, Bounded rationality, Computer economics.**

I. Introduction

THE complexity of the analytical analysis of financial markets is the main cause for the use of other alternative methodologies to gain a better understanding of some of the unsolved problems in finance. Despite the existence of previous or contemporary works, the most influential work in artificial financial markets is the Santa Fe Artificial Stock Market [45], [3]. A good introduction to Computational Finance can be found in [52], to Agent-based Financial Markets in [28] and to Agent-based Computational Economics in [51].

This branch of research is inspired in the notion that financial markets can be seen as an adaptive complex system in which rich dynamics exists and is full of emergent properties. Such rich dynamics and emergent properties should arise endogenously rather than being imposed exogenously. By using this approach, the intention is to overcome the limitations of the traditional theory in which many unrealistic assumptions have to be made to allow analytical tractability.

Artificial financial markets of all sorts and flavors have been developed in the last decade and are still being created with an always increasing complexity and proximity to reality that was not possible in the past. The research in this field is now mature and its acceptance in Economics has finally increased. This area has witnessed a sustained increase in the number of papers published related to this field.

Although they all differ in the sort of assumptions made, the methodology and tools used; these markets share the same essence: the macro behavior of such market (usually the price) should emerge endogenously as a result of the micro-interactions of the (heterogeneous) market participants. This approach is in opposition with the traditional techniques being use in Economics and Finance.

Our approach to the modeling of artificial stock markets is different to the above mentioned cases mainly on the strategic behaviour of the agents. We use a very simple market mechanism and sophisticated agents, because our aim is to study the co-evolution of the group of genetic programming based agents and the consequences on the price of changes in the agents' strategic behavior.

We are interested as well in finding the conditions under which the statistical behaviour of the endogenously generated price resembles the behaviour of real prices. The market reported in this work is composed by different types of traders: technical traders, fundamental traders, and noise traders. The market mechanism, the agents' strategic behavior and the relevant parameters will be described in detail in later sections. Additionally, with the purpose of investigating the role of heterogeneity in artificial financial markets, we have developed a flexible software platform with sophisticated traders and an evolutionary inspired constraint.

A. Statistical properties of stock returns

The statistical analysis of the price time series is usually performed on the continuously compounded return or log return. The log returns are defined in the following way:

$$r_t \equiv \log \frac{P_t}{P_{t-1}} = p_t - p_{t-1} \tag{1}$$

where $p_t \equiv \log P_t$. Some of the advantages of such returns are first that the continuously compound multiperiod return is the sum of continuously compounded single period returns, and second that it is more easy to derive the time-series properties of additive processes than multiplicative processes.

Time series of stock returns exhibit interesting statistical features which seem to be common to a wide range of markets and time-periods. Such statistical properties are known as *"stylized facts"* and have been reported for several types of financial data and their presence seems to be ubiquitous in all sorts of financial markets [10], [37], [38].

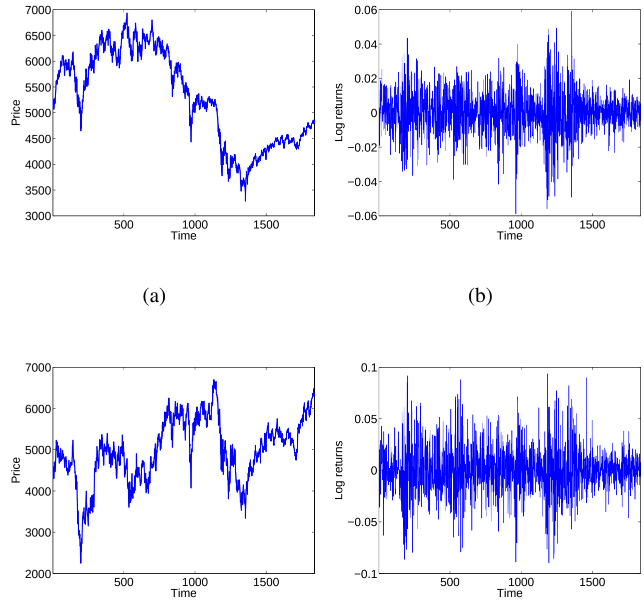
Such statistical properties of the returns have become a very important benchmark for the researchers of artificial financial markets. Such properties are the first step to accomplish when building a simulated financial market [32]. Moreover, some artificial markets try to explain the origins of such stylized facts [36].

We will not report all of such stylized facts for our experiments due to different reasons, like the frequency of our generated prices (which we will interpret as daily closing prices). Therefore, we will describe briefly, as they are described by Cont in [10], the facts that we will be reporting in later sections:

1) Lack of autocorrelations: (linear) autocorrelations of returns are usually insignificant. However, this is not true for small intra-day time scales.
2) Volatility clustering: different measures of volatility display a positive autocorrelation over several days, which quantifies the fact that high-volatility events tend to cluster in time. As noted by Mandelbrot, “large changes tend to be followed by large changes, of either sign, and small changes tend to be followed by small changes”.
3) Slow decay of autocorrelation in absolute returns: the autocorrelation function of absolute returns decays slowly as a function of the time lag, roughly as a power law with an exponent $\beta \in [0.2, 0.4]$. This is sometimes interpreted as a sign of long-range dependence.
4) Heavy tails: The distribution of daily and higher frequency returns displays a heavy tail with positive excess kurtosis. The tail index is finite, higher than two and less than five for most assets, exchange rates and indexes.
5) Conditional heavy tails: even after correcting returns for volatility clustering (e.g. via GARCH-type models), the residual time series still exhibit heavy tails. However, the tails are less heavy than in the unconditional distribution of returns.
6) Non Gaussianity: the stock returns on a weekly, daily and higher frequencies fail to be normally distributed.

Figure 1, illustrates the daily closing prices 1(a) and log returns 1(b) for the FTSE100 index and for the Barclays bank's share 1(c) and 1(d) from the 2^{nd} of January 1998 to the 31^{st} of December 2004.

In order to verify that our endogenously generated price mimics the above described statistical properties, we will perform different sorts of test. For the first described property, we will report the autocorrelations of the log returns, the absolute log returns and the squared log returns for different time lags. If the first property holds, one should observe that the log returns' autocorrelations for different lags should be around zero. In Figure 2, we can observe that the log returns' autocorrelation is effectively around zero for the FTSE100 index and the Barclays bank's share. However, we can see in the same figure, that such lack of autocorrelations does not happen for the absolute or squared log returns, which is a quantitative signature of the phenomenon known as volatility clustering (property number two). The property number three can be also verified in Figure 2, we can see there that the autocorrelation of the absolute and squared log returns decays



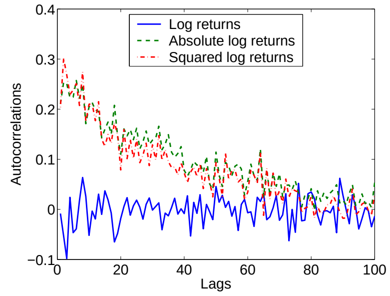
(a) (b) (c) (d)

Fig. 1. Price and log returns for the FTSE100 (a) and (b); and the Barclays bank's share (c) and (d).

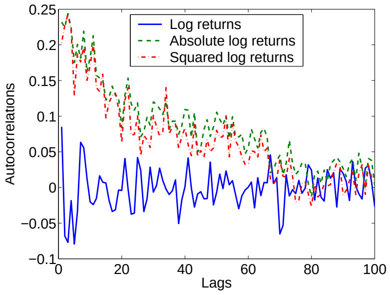
until it is practically zero for lags larger than eighty days.

The distribution of financial time series displays “fat tails”. The term “fat tails” refers to higher density on the tails of a distribution in comparison to the tails' density under the normal distribution. In order to be able to determine the shape of the tail one must estimate the shape parameter (α) or the tail index (τ). The Hill tail index ([22]) is an estimator of the α parameter and it could be considered as an standard tool for the study of tail behavior of economic data due to its good performance and simplicity. However, one of the main problems on the application of such index is that it is necessary to define a priori the size of the tail. To overcome such limitation, the fourth property is going to be tested by calculating and reporting the Hill tail index for different tail sizes (0.1%, 0.5%, 1%, 2.5%, 5%, 10% and 15%). Additionally, we will report the returns' kurtosis. For a normal distribution the kurtosis is three. However, it has been found in financial data sample kurtosis larger than three. This phenomenon is known as excess kurtosis and is an indication of fat tails.

The fifth property is going to be tested by reporting the ARCH and GARCH coefficients. Both coefficients should be less than one. The property number six is going to be tested by the the Jacque-Bera test, which indicate us if the sampled data is drawn from a Normal distribution or not. Table I-A, shows for the FTSE100 and the Barclays' share some basic statistics, the GARCH and ARCH coefficients, skewness, kurtosis, the Jacque-Bera H value, the correlation coefficient with lag one and different Hill tail indexes for various tail sizes. Sample kurtosis is an useful indicator of fat tails and of departure from normality, such statistic should be three for the Normal



(a)



(b)

Fig. 2. Autocorrelations for different lags of the log returns, absolute log returns and squared log returns on the FTSE100 (a) and on the Barclays bank's share (b).

distribution and a value larger than three indicates that the distribution possess fat tails. Typical values for sample kurtosis in exchange rates, indexes and high frequency data are much larger than three.

We can observe in Figure I-A typical values for the kurtosis (5.13829 and 4.62582) and for the Hill tail index. Depending on the percentage of extreme returns taken from the sample, such index takes values from 6.7913 to 1.94232. In most of the financial time series, the Hill tail index takes values from 5 to 2. Another interesting value to look at, is the result of the Jacque-Bera test, which in the two reported cases rejects the null hypothesis that the sampled data is drawn from a normal distribution.

II. CHASM

The Co-evolutionary Heterogeneous Artificial Stock Market (CHASM) can be considered as a software platform that allows the user to perform a series of experiments that contemplate different aspects of our simulated financial market. In this section we explain the abstract model and its main characteristics, later we explain the software and its interfaces.

A. Overview of the model

The market is populated by traders that will interact with each other by means of buying and selling some assets. The market participant i will be able to hold at time t, two different types of assets:

Statistics	FTSE100	Barclays
Mean	-0.0000341	0.0002035
Median	0	0
Minimum	-0.058853	-0.089806
Maximum	0.059026	0.093740
Std. Dev.	0.012459	0.023342
GARCH coefficient	0.89999	0.899536
ARCH coefficient	0.089564	0.089988
Skewness	-0.133266	0.113409
Kurtosis	5.13829	4.62582
J-B Test H value	1	1
Corr. coefficient	-0.008316	0.085004
AlphaHill 1 %	5.05533	6.7913
AlphaHill 2.5 %	3.96377	4.89301
AlphaHill 5 %	3.24536	3.35794
AlphaHill 10 %	2.61432	2.26481
AlphaHill 15 %	1.97705	1.94232

TABLE I

STATISTICS FOR THE LOG RETURNS FTSE100 AND BARCLAYS.

- a risky asset, denoted by $h_i(t)$ or
- cash, denoted by $c_i(t)$

The market is composed of technical, fundamental and noise traders. We define N_T as the number of technical traders, N_F as the number of fundamental traders, N_N as the number of noise traders and N as the total number of traders in the market. The stock price at time t will be denoted by $P(t)$.

At the beginning, all the agents are endowed with a certain number of shares and a certain quantity of cash, both specified by the investigator. Their position on each of the assets might change as a result of the agents' decision to sell or buy a certain quantity of the risky asset.

B. Market mechanism

The market mechanism that we use in this paper is similar to the one introduced in [20]. We chose a simple mechanism in order to avoid the complexity that could prevent us from understanding the important aspects that would lead to the reproduction of the statistical properties of real financial markets.

Our interest is on the impact that the agents change in behavior, or the differences in information, or the differences in computational capabilities, or the generation of limit orders have on the price and on the traders wealth distribution.

The participants will take a decision $d_i(t)$ at each time step of the simulation. We denote the fact that an agent takes a decision to buy by $d_i(t) = 1$, to sell by $d_i(t) = -1$ or to do nothing by $d_i(t) = 0$. Moreover, they will make a bid or offer of just a fraction $q_i(t)$ of their current holdings in the following way:

$$q_i(t) = \begin{cases} g\frac{c_i(t)}{P(t)} & if \quad d_i(t) = 1 \\ -gh_i(t) & if \quad d_i(t) = -1 \\ 0 & if \quad d_i(t) = 0 \end{cases} \tag{2}$$

The fraction g of change in the agents' holdings is an important parameter of our simulation and is related to the cautiousness of the agents.

The aggregated volume of bids, denoted by $B(t)$ and the aggregated volume of offers, denoted by $O(t)$ will be used to calculate the excess demand $D(t) = B(t) - O(t)$

To determine the price based on the excess demand $D(t)$, we follow a price determination equation similar to the one in [17], [11], and [23]. The price is then calculated in the following way:

$$P(t) = P(t-1) + D(t)/\lambda \tag{3}$$

The parameter λ represents the sensitivity of the market to the order imbalance. In our market λ was set to be the number of traders participating in that trading round times a constant c, which is yet another parameter of our simulation.

The orders placed by the different types of traders will be interpreted as market orders. However, the technical traders will be able to place limit orders in particular situations to be explained later. Such market and limit orders will be just partially satisfied. The fraction of unsatisfied orders will be the same for all the agents.

The rationing of the fulfilled orders is similar to the one introduced in [20]. The total number of sell orders is $O(t)$ and the total number of shares that can be bought at the new price can be calculated as follows:

$$\tilde{B}(t) = B(t)\frac{P(t-1)}{P(t)} \tag{4}$$

The fraction of filled buy δ_+ and sell orders δ_- can be described as follows:

$$\delta_+ = \min\left(1, \frac{O(t)}{\tilde{B}(t)}\right) \quad , \quad \delta_- = \min\left(1, \frac{\tilde{B}(t)}{O(t)}\right) \tag{5}$$

Having this, we can now calculate the amount of shares that the agent i will buy or sell $\rho_i(t)$ as:

$$\rho_i(t) = \begin{cases} g\delta_+\frac{c_i(t)}{P(t)} & if \quad d_i(t) = 1 \\ -g\delta_- h_i(t) & if \quad d_i(t) = -1 \end{cases} \tag{6}$$

Finally we can update the traders holdings of cash and the risky asset:

$$\begin{aligned} h_i(t) &= h_i(t-1) + \rho_i(t) \\ c_i(t) &= c_i(t-1) + \rho_i(t)P(t) \end{aligned} \tag{7}$$

C. Traders

Despite the complex task of defining all the different sorts of traders that intervene in a financial market, there are some well accepted classes of traders that are commonly used on the literature. In this work we will limit such classes of traders to three different types: technical traders, fundamental traders, and noise traders. None of our traders follow rational expectations and their interaction will be only by means of the price. In the Artificial Financial Markets literature we will find mostly those three types of traders, although the specific mechanisms and implementations can vary widely.

1) Noise traders: The noise traders will take a decision to buy, sell or do nothing with different probabilities p_b , p_s and p_n respectively. Such probabilities are defined before the simulation and remain with the same value during the simulation. This type of traders were included to represent a justifiable source of noise.

2) Value Traders: The behavior of the fundamental traders is taken from [17]. The basic idea behind the strategy of such traders is that they will change their position on the risky asset if the price departs from a value that they perceive as the fundamental one. These traders will continue to adjust their positions until such difference T, is lower than a certain threshold value τ.

The afore mentioned values will be generated for each individual trader by drawing random numbers from uniform intervals $[T_{min}, T_{max}]$ for T and $[\tau_{min}, \tau_{max}]$ for τ. The limits of such intervals represent another four parameters of the simulation.

3) Technical traders: We consider technical analysis as a key feature for the modelling of the behavior of this group of agents, despite the open debate between academics and practitioners about it. We believe that technical analysis is an important tool for decision making in investment [7]. Besides, there is strong evidence that technical analysis is being used extensively in financial markets.

Financial forecasting using neural networks, genetic algorithms ([21]), genetic programming ([26]) and other machine learning techniques has been a very dynamic field of study. We can see some important work in the nineties and some innovative recent proposals [1], [4], [5], [43], [12], [13], [18], [19], [53].

Broadly speaking, the technical traders in our artificial market forecast if the price is going to rise by a certain $r\%$ within a certain n number of days. For that purpose, they will be equipped with up to twelve different technical and momentum indicators to form investment decision rules. Such indicators are well known technical indicators. We chose a short period indicator and a long period one because it is the way in which they are used by technical analysts. We will give more detail regarding the indicators in subsection II-D.3.

Each technical trader owns a population of such investment decision rules, represented by decision trees. The decision trees are randomly initialized and then evolved by an evolutionary mechanism known as Genetic Programming (GP). EDDIE ([53], [33], [54], [40], [56]) constitutes the basic platform to the design of the investment strategy of this group of agents. The selection of genetic programming was mainly due to the fact that it has been used previously to model agents in artificial financial markets [14], [15], [58], [8] and [41], [42].

It is important to point out that the agents could use any kind of financial information. However, we did not want to have many exogenous sources of information and we decided to let the traders built their own decision rules with technical and momentum indicators, considering that we were able to generate them all just by using the price. Additionally, under certain circumstances they will be able to behave like the fundamentalists.

This group of traders is the richest in behavior and the most complex of the market participants. Such traders are organized in heterogeneous groups and the agents inside each group share all the same parameters. The heterogeneity among the groups will come from several sources: generation of limit orders, computational capability, information, time horizon and desired rate of return. We will explain in detail such sources of heterogeneity later in this section.

The agents inside a group share the following parameters and characteristics:

- Technical indicators
- Logical functions
- Relational functions
- Genetic Programming parameters
- Time horizon and desired rate of return
- Retraining condition
- Limit orders generation
- Fundamental behavior.

D. Important Features in CHASM

In this section we will describe in detail some of the most important features of CHASM. The implications of changes in such important aspects of our model will be described by experimentation in the next section.

1) Market and limit orders: When a person, a professional trader, a market maker or a corporation are trading on a stock market, there are different ways to do so. After the decision making process of any of such entities an order must be submitted to a broker (or the representant that is trading on behalf of them). Essentially [1], there are two main types of orders:

- Market orders
- Limit orders

The market orders are buying or selling orders that must be executed at the current price of the stock on the market. There is certainty about the execution of a market order but uncertainty about the execution price. On the other hand, limit orders are buying or selling orders in which the trader specifies the price at which she is willing to trade (such prices are called bid or ask prices). In the case of limit orders, there is certainty about the execution price, but there is no certainty about the execution of the order.

In order to have a complete and realistic investment strategy, we incorporated certain types of limit orders in addition to the market orders to model an exit strategy for the agents. We have basically two types of limit orders: profit taking limit orders and stop loss limit orders.

The profit taking limit orders are orders for selling that are sent after a purchase of a stock takes place. The asking price for such limit orders must be higher than the purchasing price that originated the profit taking limit order. The basic idea behind such order is to lock a certain profit realization for the trader. However, there is no certainty of the execution of such type of orders.

[1]there is a number of other types of orders, but it is not our intention to give a full account of them

The stop loss limit orders are orders for selling a stock that is being held by an investor. Such order becomes a market order after the price is at or goes below a threshold price (stop price) defined by the investor. The basic idea behind such limit orders is to try to limit the loss of an investment made by the trader.

The profit taking limit orders are incorporated in our model to provide the agents' with a complete investment strategy. To be more specific, if the agent forecasts that the price is going to rise and he buys now a certain amount of shares, he would have to sell them whenever the price actually reaches such forecasted increase. On the other hand, if the price does not rise beyond her forecast she should sell at the end of her time horizon due to budget constraints.

The stop loss limit orders are incorporated as well as part of an exit strategy for our agents. The generation of such orders and the profit taking ones during the trading has important repercussions on the dynamics and the statistical properties of the price as we will see on the experiments reported in the next section.

2) Fundamental trading: In addition to the incorporation into the market of fundamental traders, CHASM allows us to incorporate fundamental like behavior on top of the technical traders. This characteristic of the technical traders in our model can be justified by arguing that in real life some traders do use technical analysis in conjunction with fundamental analysis. These traders know that the price of a certain stock is well beyond a reasonable value (fundamental value); however, they still follow the trend a little longer (short time horizon) in order to make a profit out of it. In [50] the authors report that more than 90 percent of dealers in the foreign exchange market use some form of technical analysis and in short time horizons, technical analysis predominates over fundamental analysis.

In CHASM, we are able to have technical (fundamental) traders that behave like fundamental (technical) traders under certain specific circumstances. These traders will behave like technical traders until the price is well beyond a reasonable (fundamental) value. Then, they trigger the fundamental trading until such discrepancy disappears. We will report the experimental implications of such behavior later in this work.

3) Indicators: The indicators used by the technical traders to forecast increases or decreases in the price are a very important aspect of our market. Such indicators can make a substantial difference on the behavior of the endogenously generated price.

The indicators used for the current work consist of technical, momentum and volatility indicators. The different indicators that were used and their periods are: The price moving average of the last 12 and 50 days, the trading breakout rule of the last 5 and 50 days, the filter rule of the last 5 and 63 days, the price volatility of the last 12 and 50 days, the momentum of the last 10 and 60 days and the momentum moving average of the last 10 and 60 days. We use a short horizon and a long horizon indicator because that is the way in which they are used by practitioners of technical analysis.

Given a price time series $\{P(t), t \geq 0\}$ and given a period of length L, we will define our interpretation of some popular technical indicators as follows:

The Moving Average indicator is defined as:

$$MA(L,t) = \frac{P(t) - \left(\frac{1}{L}\sum_{i=1}^{L} P(t-i)\right)}{\frac{1}{L}\sum_{i=1}^{L} P(t-i)} \quad (8)$$

The Trading Breakout indicator is defined as:

$$TRB(L,t) = \frac{P(t) - max\left\{P(t-1), ..., P(t-L)\right\}}{max\left\{P(t-1), ..., P(t-L)\right\}} \quad (9)$$

The Filter indicator is defined as:

$$Filter(L,t) = \frac{P(t) - min\left\{P(t-1), ..., P(t-L)\right\}}{min\left\{P(t-1), ..., P(t-L)\right\}} \quad (10)$$

The Volatility indicator is defined as:

$$Vol(L,t) = \frac{\sigma\left(P(t), ..., P(t-L+1)\right)}{\frac{1}{L}\sum_{i=1}^{L} P(t-i)} \quad (11)$$

where σ, represents the standard deviation.
The Momentum indicator is defined as:

$$Mom(L,t) = P(t) - P(t-L) \quad (12)$$

The Moving Average Momentum indicator is defined as:

$$MomMA(L,t) = \frac{1}{L}\sum_{i=1}^{L} Mom(L, t-i) \quad (13)$$

We chose such indicators mainly because they proved to be useful on forecasting rises and drops of the price in previous works like [7], [53], [33], [54], [55], [56], [18] and [19]. However, due to the design of the forecasting mechanism, there is no reason to stop us from using some other information like more sophisticated technical indicators, information from the limit order book, market microstructure information, fundamental information, etc. Additionally, we performed a sort of standardization in order to avoid that the range of numbers, generated by the GP mechanism, could very large and therefore increasing the size of the search space.

Further in this work we will explore experimentally the implications of changes on the information (modeled here by the above listed indicators) provided to the different groups of technical traders.

4) Desired return and time horizon: The technical traders will be organized in groups that will share some common characteristics, as it has been described previously. Among such characteristics we have the desired rate of return and the corresponding time horizon to achieve such rate of return. These two characteristics (parameters) proved to be of central importance on the behavior of the market, since our GP agents work as classifiers.

The majority of the simulated markets possesses agents that do not consider a multi-period preferences and the agents share the same planning, forecasting and decision making horizon [28]. CHASM is different to most of the previously designed models in the forecasting mechanism and the heterogeneity of time horizons.

The GP forecasting mechanism of our agents works by classifying the training cases in three different classes: buy, sell or hold. Such classification depends on the time horizon provided as the mechanism will verify for each data point if in the near future (time horizon) effectively, there was a rise (drop) on the price by more (less) than the desired rate of return.

If the selection of such quantities is unreasonable, it will cause the classifier to be biased towards a certain class. Therefore, creating unrealistic (unreasonable) investment rules and having an unrealistic price and statistical properties of the returns as a result. We designed some experiments to test the impact that this feature of CHASM has on the dynamics of the simulation.

5) Trading proportion: The trading proportion is a parameter of the market that controls the proportion of the asset or cash that the traders would commit on each of the operations that they will perform during the trading rounds as it can be clearly seen in 2. Such trading proportion is a quantity that we can use to model the degree of cautiousness of the agents in our market.

The trading proportion proved to be of central importance in our simulations. The implications of this important feature of our market will be tested experimentally and described in the next section.

6) Fitness function: The fitness function is a very important aspect of our market, we use a prediction accuracy fitness function to drive the GP mechanisms of each of the agents. The fitness function used as a rate of accuracy was the rate of correctness defined as the number of correct classification over the total number of cases. There exist the possibility of using some other ingredients (like the rate of failure) on the fitness functions to bias the search over the solution space. Some interesting changes to the fitness measure have been implemented and tested in [34].

The flexibility on the manipulation of the fitness function due to the original design of the GP forecasting mechanism is one of the main advantages that we consider important in the use of GP as a successful technique to perform financial forecasting.

III. Learning to forecast investment opportunities

In this section we explain the main forecasting mechanism which is the basic framework for the decision making process of the technical traders. Genetic programming is at the heart of such mechanism and it has been used in the past to perform technical analysis by several research groups like [43], [12], [19].

The modeling of the learning process by the agents is a central part of our research agenda. Regarding the agents' learning process, we consider of extreme importance what Lucas wrote in [35]:

> In general terms, we view or model an individual as a collection of decision rules (rules that dictate the action to be taken in given situations) and a set of preferences used to evaluate the outcomes arising from particular situation-action combinations. These decision rules are continuously under review and revision; new decision rules are tried and tested against experience, and rules that produce desirable outcomes supplant those that do not.

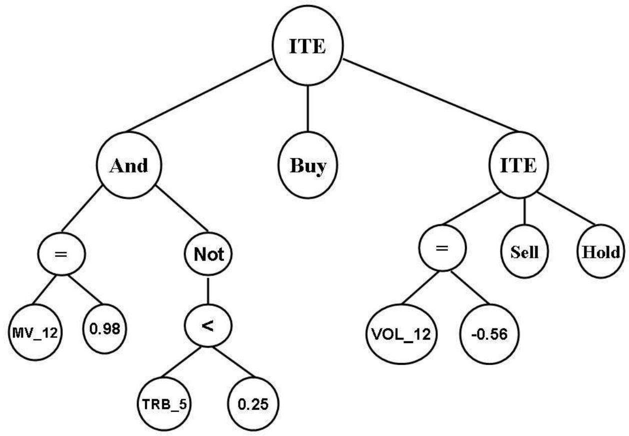


Fig. 3. Example of a decision tree.

For the modeling of the learning process described above by Lucas we will use genetic programming. Such technique has been previously described as a suitable way to model economic learning in [6]. The learning process that we used to model our agents' behaviour will be described further in this section.

A. Forecasting with EDDIE

We use the architecture for EDDIE explained in [53] and [33] for the elaboration of the agents' decision rules which recommend whether to buy, hold or sell. As is standard with genetic programming, each agent is assigned an initial population of decision rules randomly generated. These include well known fundamentals based forecasting rules or trend following moving average type technical rules. Candidate individuals are selected randomly, biased by their fitness, for involvement in generating members of the next generation. General mechanisms (referred to as *genetic operators*, e.g. selection, crossover, mutation) are used to combine or change the selected candidate individuals to generate offspring, which will form the population in the next generation.

In EDDIE, an individual is represented by a decision tree. The basic elements of such decision tress are *rules* and *forecast values*. A single rule consists of one useful indicator for prediction, one relational operator such as "greater than", or "less than", etc, and a threshold (real value). Such a single rule interacts with other rules in one decision tree through logic operators such as "Or", "And", "Not", and "If-Then-Else". Forecast values in this model are directions of price movements, either a positive trend (i.e. positive x% return within specified time interval can be achievable) or negative trend (i.e. negative x% return within a specified time interval can be achievable).

Figure III-A shows and example of one possible decision tree. In such figure we can see that the root node is always an If-Then-Else node, the left child an If-Then-Else node is a "condition" node. Additionally, there are two right children which could be either a "decision" node or another If-Then-Else node. The rule that is being represented by the decision tree in Figure III-A is the following one:

If $((MV\_12 = 0.98) AND (NOT(TRB\_5 < 0.25)))$ **Then**
    Buy
Else
    If$(VOL\_12 = -0.56)$ **Then**
        Sell
    Else
        Hold
    End if
End if

Fig. 4. Example of a decision rule interpreted from a decision tree

The type of node is going to become relevant when applying the crossover and mutation genetic operators because it is precise to maintain the tree consistency. For example, in the case that a mutation operation is going to take place and the selected node where such mutation is going to happen is a "condition" node; then, the randomly generated mutation must be a "condition" like node. The same should happen for the crossover operation: there must be compatibility between the subtrees than are going to be exchanged by the parents.

Recommendation to BUY at t follows from the prediction of a price rise (positive trend) over a given period, recommendation to do nothing (HOLD) follows from the fact that there is no evidence of a price rise or a price drop and recommendation to SELL follows from the prediction of a price fall (x% negative trend). Note different returns thresholds and horizons exist for different classes of traders. Since decision trees are used to predict directions of price changes and make recommendations for trade, the success or failure of recommendations can be categorized as a three-class classification problem. Each prediction point for every decision tree can be classified into either a positive position, a holding position or a negative position. For each decision tree, we define RC (Rate of Correctness), and RF (Rate of Failure) as its prediction performance criteria. Formula for each criterion is given through a contingency table in Table II as follows:

Lets define: RC as the Rate of Correctness; and RF as the Rate of Failure.

TABLE II

A CONTINGENCY TABLE FOR THREE-CLASS CLASSIFICATION/PREDICTION PROBLEM

	Predicted price rise (PBs) **BUY**	**Predicted no inf.** (PHs) **HOLD**	**Predicted price drop** (PSs) **SELL**
Actual price rise (ABs) **BUY**	# of True Buys (TB)	# of Actual Buy Predicted Hold (BH)	# Actual Buy Predicted Sell (BS)
Actual no inf. (AHs) **HOLD**	# of Actual Hold Predicted Buy (HB)	# of True Holds (TH)	# of Actual Hold Predicted Sell (HS)
Actual price drop (ASs) **SELL**	# of Actual Sell Predicted Buy (SB)	# of Actual Sell Predicted Hold (SH)	# of True Sells (TS)

$$RC = \frac{TB + TH + TS}{ABs + AHs + ASs} \tag{14}$$

$$RF = \frac{HB + SB}{PBs} + \frac{BH + SH}{PHs} + \frac{BS + HS}{PSs} \tag{15}$$

Each agent selects the decision tree which constitute trading strategy to buy or sell that maximizes the fitness function

$$\Gamma_{(1)} = \varphi(rc)RC - \varphi(rf)RF \tag{16}$$

The fitness function involves two performance values, i.e. RC and RF, each of which is assigned a different weight $\varphi(rc)$ or $\varphi(rf)$ respectively. While the fitness function can guard against loss making positions, the population of decision trees from which agents conduct their search may lead to investment income under-performance. One important advantage of genetic programming is that we can bias the search mechanism by using different values for such weights. However, we used a value of zero for the weight $\varphi(rf)$. In other words, we used just the rate of correctness as the performance criteria to drive the evolutionary mechanism.

We implemented a very efficient method to avoid bloat and to speed up our simulation, see [47]. Bloat happens during the evolutionary process, when the trees grow in size but there is no improvement in fitness, see [27]. Essentially, this means that there are some branches of the trees that are redundant or even worse, they reduce the fitness of the individual.

We considered that the control of bloat is important because it was necessary for us to generate realistic investment rules. It would be very difficult for us (and for anyone) to justify very complex rules being used by the agents.

B. Learning

Learning is a key factor in our simulation and as we were investigating the repercussions of the type of the learning frequency on the endogenously generated price, we had to verify that the learning process has been beneficial. To be more precise, learning should enable the individual to improve her wealth in relation to the other traders. The technical traders are the only type of traders that will be able to learn during the simulation of the market.

To investigate this issue, we performed several experiments in the following way: on each experiment we replicated one trader (technical) and followed both traders' wealth (the original and the replicated one) throughout the whole simulation. We set the replicated trader to be able to retrain every 1000 steps of the simulation. We can see clearly in Figure III-B that after the execution of the GP mechanism with the endogenously generated price, the replicated trader improved her wealth in comparison to the original trader. We executed these experiments several times with different traders from different groups and in the vast majority of the cases the replicated traders did better than the original ones.

We allowed the replicated trader to retrain every 1000 periods of trading because we wanted to give the original trader the opportunity to improve during that period and may be even to perform better than the retrained agent, as we can see in Figure 5(a). Additionally, we wanted to avoid that too many things changed at the same time and that the comparison became impossible. We could certainly try to perform experiments with some reasonable shorter lapses of time, but the computational cost plays a role as well.

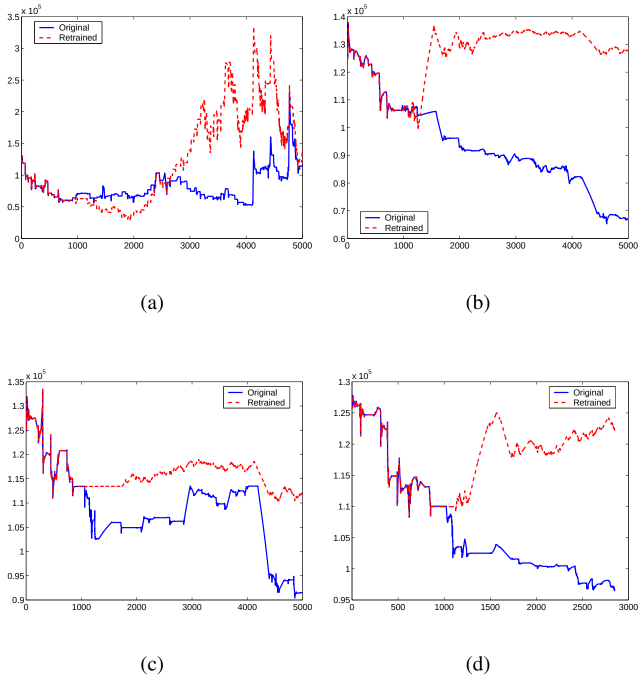


Fig. 5. Examples of wealth evolution with and without learning.

During the retraining phase, the GP mechanism is executed with the same conditions as it was executed for the initial training. This is done with the objective of preserving what we perceive as a realistic and competent forecasting mechanism. It is worth mention that the rate of correctness of the agents during the initial trading is above sixty percent.

It is important to point out that in our case, the fitness measure used to drive the evolution process was the rate of correctness. Moreover, such classification measure can be translated into an improvement on the traders wealth. We consider such result as an important one in our research that can be stated in the following way: a classification rate driving the evolutionary process has a direct impact on the agent's wealth.

IV. The Simulation

The market will operate as if each trading round is one day. This is due to the fact that our technical traders were trained with daily closing prices. However, there is nothing to prevent us from interpreting the time in another scale or train the agents with high frequency data. The market participants will be able to trade on every single round of the market with some exceptions to be explained next.

Noise traders will take a decision to buy, sell or do nothing based on the probabilities assigned for each decision. They will be able to participate in the market on every single iteration of the simulation program.

The fundamentalists enter on a buying (selling) cycle if there is a difference between the stock's price and the fundamental value beyond certain threshold value T. They will stay in such cycle until the difference is smaller than another threshold value τ. After the return of the stock's price to the fundamental value, this type of traders will review again if there exists a difference between the price and fundamental value and so on.

The technical traders can change their position on each trading round based on their forecasts, unless they have some pending limit orders to execute. Once they have completed the round trip (the investment decision and the exit strategy), they can take another decision based on the generated rules.

Once all the market participants make their bids and offers, we calculate the excess demand and then the price can be updated. Afterwards, each agents' orders are partially satisfied by a proportion that clears the market considering the new price. Finally, the holdings of the risky asset and cash are updated for each of the traders that participated in the trading round.

After all the above steps are executed, each technical trader reviews its retraining condition. In the model there exist two types of conditions for retraining: in fixed time intervals and in an endogenous way known as *Red Queen* retraining.

In the case that the retraining periodicity is set to be in fixed time intervals, the trader launches the GP mechanism creating the initial population of rules with half of her current population and the other half randomly generated.

The case of the Red Queen retraining is more complex: retraining for a certain agent will take place whenever the agent's wealth falls below the average wealth. The agent's initial population for each retraining process is generated in the same way as was described for the retraining with fixed periodicity.

A. Parameters

We have created a flexible model in which we have a large number of different parameters for us to explore and analyze different phenomena in financial markets. These parameters are going to be divided mainly in two different classes: market parameters and traders' parameters.

1) Market Parameters: The market parameters control some of the general parameters of the simulation, they determine for example the proportion to invest, the number of periods of trading, etc. Below is a list of the relevant market parameters:

- Number of trading periods
- Proportion to trade
- Price constants
- Random seed

2) Traders' parameters: The different types of traders possess different parameters. For example, the noise traders only have three parameters: the probability to buy, the probability to sell and the probability to do nothing. On the other hand, the fundamentalists have four constants as parameters: T_{min}, T_{max}, τ_{min} and τ_{max}. All the different traders share as parameters the initial number of shares and the initial amount of cash available at the beginning of the simulation.

The group of technical traders is organized in several groups. Within such groups they share the same parameters. This organization allowed us to model a key factor in our research: heterogeneity. It is possible to split the parameters of the technical traders in three different types: group parameters, information parameters and genetic programming parameters.

The most important group parameters are:

- Desired rate of return
- Time horizon
- Memory length
- Retraining condition
- Fixed retraining periodicity
- Fitness function type
- Limit order to sell generation
- Limit order to buy generation
- Fundamental behaviour
- T_{min}, T_{max}, τ_{min} and τ_{max}

The information parameters refer to the technical indicators that the agents can use to generate the investment rules during the execution of the GP mechanism. With the parametrization of the indicators we can model different types of technical traders. For example, we can have moving average or momentum traders.

Within the parameters of the technical traders, the GP ones constitute an important set for the simulation. Such parameters will determine how close to a competent trader the agents are. For example, if the traders are equipped with a limited population size and a small number of generations, there is little chance that the traders will create accurate investment rules. The most important parameters that control the genetic programming mechanism are:

- Population size
- Number of generations
- Mutation rate
- Crossover rate
- Initial tree depth
- Maximum tree depth
- Tournament size
- Probability of selecting the best individual of the tournament
- Tarpeian constant

V. Experiments

This section describes the experimental results performed in order to obtain a realistic price behavior. We seek to discover the minimal conditions under which stylized facts [2] arise in CHASM. It is worth emphasizing that with a complex model it is not a trivial task to search for such conditions. Furthermore, it is the first work (to our knowledge) in which an exhaustive search for the minimal set of conditions under which realistic price dynamics emerge. The complexity of some artificial markets usually prevents the researcher from knowing which aspect of her model is the responsible for the emergence of stylized facts.

[2]The so called "stylized facts" appear to be ubiquitous in different markets see [10] for an introduction.

Regarding the exploration of the parameters and main features, we performed a full exploration of each of the parameters involved on the simulation. For example, in the case of the trading proportion parameter we scaled it from one percent to one hundred percent. We presented the most relevant examples for illustrative purposes. In the case of the limit orders, their exploration was easier as there are only eight possible variations on this feature. In the case of the fundamental behaviour we turned on and off for every single group of the technical traders. We proceeded on the same fashion for all the above mentioned parameters and features. Nevertheless, we will only present the results of the most relevant cases.

A. Parameters and features exploration

In the previous section we described the most important features and parameters of CHASM. Due to the different possible combinations of such features and parameters, a systematic approach was necessary to discover promising areas under which we could obtain the desirable properties of the price.

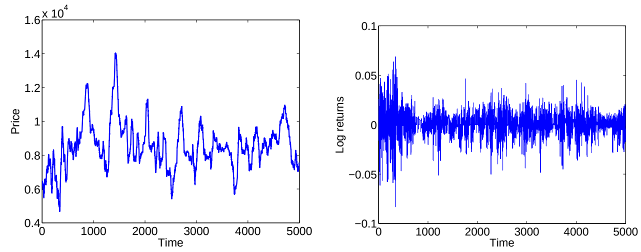
In the following subsections of this section we will first describe the characteristics of what we call the Base Case and then, we will describe the experimental results of changes on the other features and parameters of the model.

B. The base case

A large amount (one for each trader) of controlled experiments were conducted before obtaining a price that resembles the dynamics present in real prices. In this section, we present the results of a parameter setting which reproduces statistical properties of stock returns. This setting will be used as a base case for studying the effects of changing the model along individual dimensions listed in the previous section. The Base Case has the following parameters and characteristics:

- Seven different groups of technical traders.
- The groups have different indicators.
- The groups share the same desired rate of return (5.5%) and time horizon (14 days).
- The agents trade 8% of their current holdings or use the 8% of their cash to buy more shares.
- The agents generate both types of limit orders.
- The groups have the same computing power.
- There is no learning taking place.
- Group number seven of technical traders behave like value traders under certain circumstances.

Besides the set of indicators used by each group of traders and the fundamental behavior exhibited just by one group, the remaining conditions are the same for all the different groups. The indicators were assigned in the following way: group one was assigned with the two moving average indicators, group two was able to use the trading breakout indicators, group three had the filter indicators, group four used the volatility, group five used the momentum indicator, group six used a moving average of the momentum indicator and finally group seven used all the indicators. For the experiments reported in this paper the number of agents in each groups is typically three.



(a) (b)

Fig. 6. Price and log returns for the Base Case.

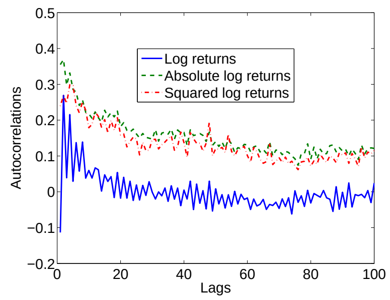


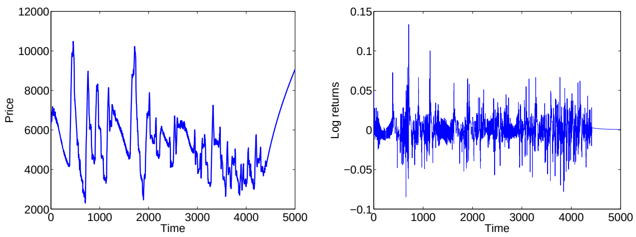
Fig. 7. Autocorrelations for different lags of the log returns, absolute log returns and squared log returns on the Base Case.

In Figure 6 we can see the price and the log returns of the Base Case. We can see that the price resembles the dynamics of the prices in real markets and the log returns capture the well known phenomena of volatility clustering. Such phenomenon can be investigated quantitatively and in Figure 7 it is possible to see the autocorrelation for different lags of the log returns, the absolute log returns and the squared log returns.

We can observe in Figure 7, that the autocorrelation of log returns is around zero, as it should be. Additionally, we can see in the same figure that for the absolute and squared log returns, there is a positive autocorrelation that decays slowly but remains positive even for lags larger than eighty. However, such positive autocorrelation is never close to zero as we saw on the cases of the FTSE100 and the Barclays bank's share.

C. Limit Orders

After the first attempts that we made at the beginning of our research, we decided that a more complete strategy was necessary to obtain realistic price dynamics. Our agents were equipped with a powerful forecasting mechanism; nevertheless, after forecasting that the price would rise and buy some assets they did not realized the profits that they would get as a reward for their forecasting ability. For example, it would be unreasonable to buy some shares today if we forecasted that the price was going to rise and later on, keep them without realizing the profit that we could get.



(a) (b)

Fig. 8. Price and log returns for the Base Case without limit orders.

Fig. 9. Autocorrelations for different lags of the log returns, absolute log returns and squared log returns on the Base Case without limit orders being generated.

The previous reason is the main cause for including an exit strategy for our agents in CHASM. Such exit strategy will be modeled by two different types of limit orders: profit-taking limit orders and stop-loss limit orders.

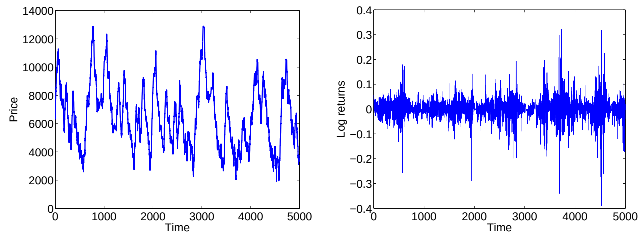
In order to test the relevance and the impact of the inclusion of limit orders on the agents' trading strategy we designed some experiments in which we turned on and off the use of one or both types of limit orders. We departed from the base case by no generating both limit orders everything else remaining the same.

In Figure 8 we can see the price and the log returns of the Base Case without limit orders being generated. We can observe by simple inspection from such figure that the price dynamics are not realistic and that such orders represent an important element in the generation of dynamics close to the reality. Moreover, at the end of the price's graph we can see that there was a sort of consensus between the traders and the price started to increase without reversion taking place. This phenomenon might be caused by the lack of a complete investment strategy as it was explained before. In other words, the agents were not selling to make profits after they bought with the purpose of making some profits.

In Figure 9, we can observe the behavior of the autocorrelations for different lags of the log returns, absolute log returns and squared log returns. Considering the two examples on Subsection I-A, we can appreciate that the behaviour of the autocorrelation for the log returns is not like in those examples (oscillating around zero). Such autocorrelation is even negative for lags between sixty and ninety. The autocorrelations for the absolute and squared log returns do not do well neither, for short lags they are very high and decay. However, they remain positive, particulary the autocorrelation for the absolute log returns.

D. *Fundamental trading*

Fundamental behavior is one of the aspects that most of the Artificial Financial Markets possess. It is a very important mechanism, in our experience, in order to avoid the price to behave in a very simplistic fashion. Before including such characteristic in our market, we had prices that were either always increasing or decreasing until they collapsed. We modeled the fundamental behavior as it was designed in [17], on top of the technical trading behavior. This means that the agents will be trading like technical traders until the price departs beyond a certain threshold value from what they consider to be the fundamental value of the risky asset.



(a) (b)

Fig. 10. Price and log returns for the Base Case without fundamental trading.

In order to test the impact of triggering the fundamental behavior we performed some experiments in which we activated or deactivated such behavior in just one of the groups. Departing from the Base Case, we turned on and off the fundamental-like behavior while in group seven everything else remained the same. We did not impose such mechanism in all the groups for two reasons: first, we wanted to have certain heterogeneity and second, we wanted to prevent that the endogenously generated price would follow too closely the exogenous fundamental value.

In Figure 10 we can see the price and the log returns of the Base Case without fundamental behavior taking place in any of the groups. We can observe that the price behaves somehow in a "reasonable" way and we can see on the log returns that they capture the volatility clustering and they look "reasonable" as well. To verify that, we have to recur to Figure 11.

In Figure 11 it is possible to observe the autocorrelation for different lags of the log returns, absolute log returns and squared log returns. In such figure, we can see that the autocorrelations of the log returns are around zero (which is fine). Additionally, we have that the autocorrelations for the absolute and the squared log returns are positive for short lags and we observe a decay that takes them close to zero. However, such autocorrelations remain positive even for lags larger than ninety.

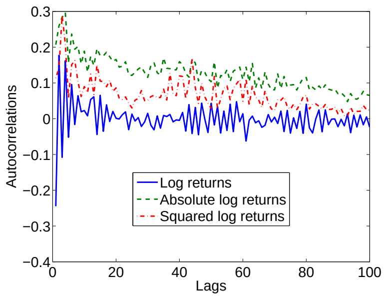


Fig. 11. Autocorrelations for different lags of the log returns, absolute log returns and squared log returns on the Base Case without fundamental behavior.

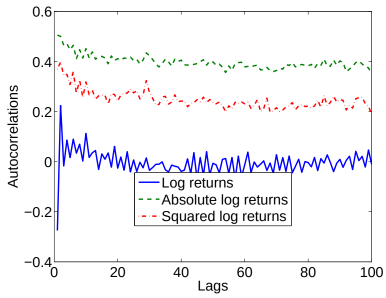
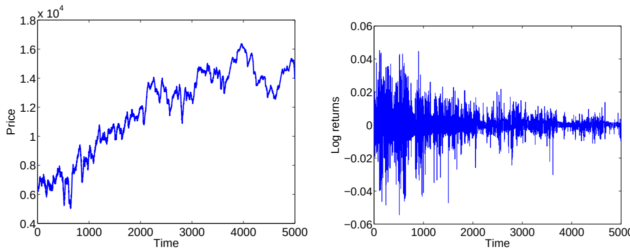


Fig. 13. Autocorrelations for different lags of the log returns, absolute log returns and squared log returns on the Base Case with homogeneous information for all groups.



(a) (b)

Fig. 12. Price and log returns for the Base Case with homogeneous indicators set for all groups.

E. Indicators

The indicator set used by each of the different groups of agents is one of the most relevant factors that we identified in order to reproduce stylized facts. We used the indicator set to model information asymmetries on the different groups of agents.

We had two different possibilities: first, as it was described for the base case, we could have that each group possess one specific type of indicator and one of the groups could possess them all; second, all the different groups could built their decision rules with all the available indicators.

We tested the two different approaches by departing from the base case and we either provided all the groups with all the available indicators or assigned the different types of indicators in the way that was described for the Base Case.

In Figure 12 we can see the price and the log returns of the Base Case with homogeneous indicators set for all groups. Despite the appearance of the endogenously generated price, we can observe that the returns do not resemble the returns present in real markets. The returns generated in our experiments for this case started to present lower volatility at later stages of the simulation. Additionally, we cannot see volatility clustering in our generated returns. In Figure 13, we can see this more clearly.

In Figure 13, we can see how the autocorrelation for the log returns does fine in comparison with the examples in SubsectionI-A. However, we observe a high autocorrelation for the squared and the absolute log returns that remains positive without decay. Therefore, this experimental setting fails to replicate the behaviour of the autocorrelation for the squared and absolute log returns.

F. Desired return and time horizon

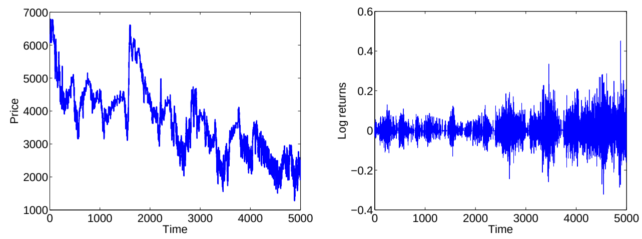
The desired rate of return and the time horizon are two very important aspects of our model. In fact, these two parameters will rule the creation of investment rules during the evolutionary process.

In order to test the impact of such parameters on the behavior of the price, we performed several experiments in which we either had homogeneity among the different groups (like in the Base Case) or we had heterogeneity in those two parameters. We have to stress the importance of a careful selection of both parameters. For example, if, for a particular group, we programmed them to ask for a large desired return on their investment, then it was unreasonable to assign them a small time horizon. Such unreasonable selection of both parameters could lead to unreasonable behavior of the agents, like agents that buy all the time or agents that do nothing the most of the time. In other words, lets assume that we want to get a 20% return, clearly this is very difficult to observe in real markets. However, if we want to achieve this in five days this is almost impossible to observe. The result of such selection would generate training data in which the class that would have the majority is the “Do Nothing” class [3]

In Figure 14 we can see the price and the log returns of the Base Case with homogeneous indicators set for all groups. In such figure we can observe an interesting price and the returns that present volatility clustering. In Figure 15, it is possible to see the autocorrelation for different lags of the log returns, absolute log returns and squared log returns. We will observe later in this section the statistical properties properties of this experimental setting. Again, what we can infer from this setting is that heterogeneity is very important to emulate the properties of real prices.

In Figure 15, we can see that despite the somehow realistic behaviour of the autocorrelations for the log returns, the autocorrelations for the absolute and squared returns is highly positive and remains in such way for all the reported lags. We

[3]Remember that the agents are trying to classify each price point to belong to the classes: “Buy”, “Sell” and “Do Nothing”



(a) (b)

Fig. 14. Base Case with heterogeneous desired return and time horizon for each group price and log returns

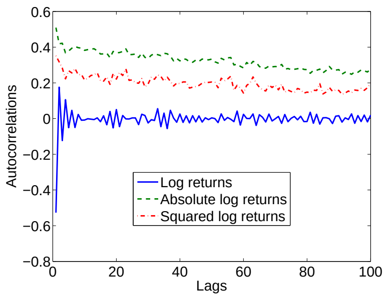


Fig. 15. Autocorrelations for different lags of the log returns, absolute log returns and squared log returns on the Base Case with heterogeneous desired rate of return and time horizon

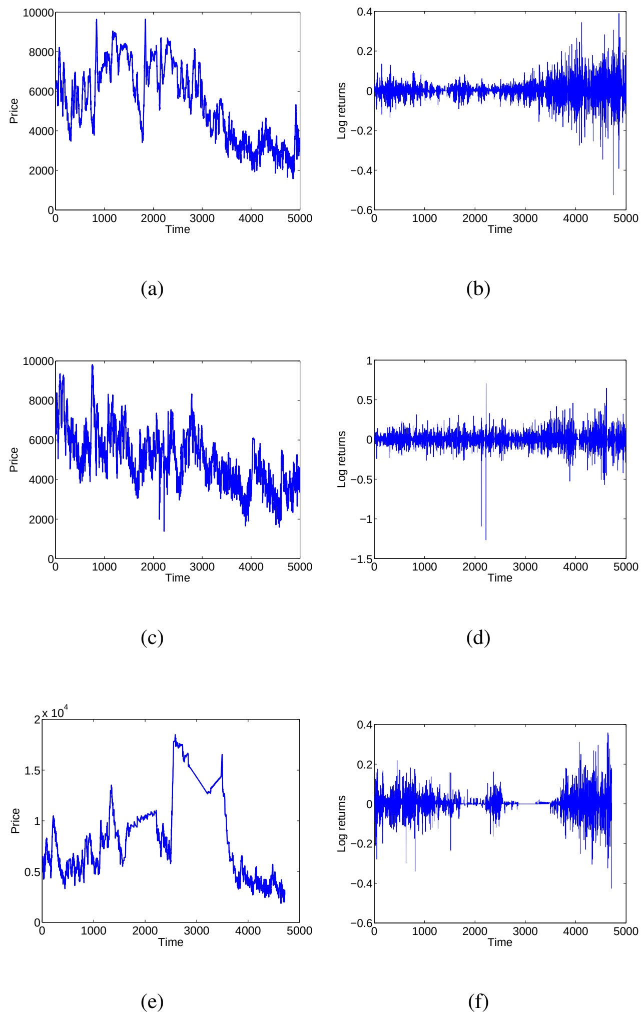
can say that we would like to observe that the autocorrelations for the nonlinear functions of the log returns to present higher decay.

G. Trading proportion

The trading proportion parameter could be used to model how cautious or aggressive to trade the agents are. If we wanted to model rather conservative agents, the trading proportion parameter would be close to 1%. On the other hand, if we wanted to model aggressive trading, such parameter would be close to 100%.

In the base case the trading proportion is 8%, from there we varied then the trading proportion parameter in order to test the impact of such parameter on the market dynamics. The changes on such parameter were essentially in one direction, this means, we increased the value of such parameter above 8% because we did not have significant changes to report with smaller trading proportions. However, we must stress that changes in such parameter were closely related to the parameter market depth on the price determination equation.

In Figure 16 we can see the price and the log returns of the Base Case with a trading proportion of 30%, 90% and 100%. Due to space limitations, we will not report the Base Case with all the other possible trading proportion values. From such figures we can see that when the trading proportion is getting close to the 100% value the dynamics worsen and the log returns can even take unreasonable values. For example, in one of the cases (90 % trading proportion) there are values



(a) (b) (c) (d) (e) (f)

Fig. 16. Base Case with a trading proportion of 30%, 90% and 100% price and log returns.

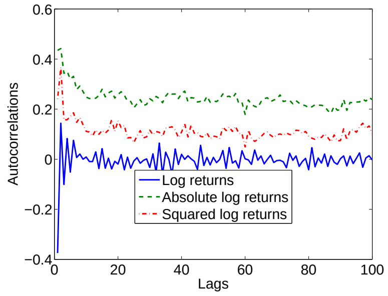
for the log return of less than -1. Such value would imply a drop on the price of more than a hundred percent!

In Figure 17, we can see the autocorrelations of the log returns, the absolute and squared log returns for different trading proportions. In Figure 17(a), we can see that the behaviour of the autocorrelations of log returns for a trading proportion of 30% is somehow acceptable. However, for the absolute and squared log returns, the decay of the autocorrelations is not enough to mimic the behaviour of real financial time series.

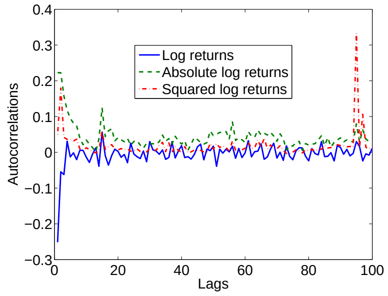
For the 90% trading proportion, the behaviour of all the autocorrelations is far from realistic (Figure 17(b)). In Figure 17(c), we can observe an acceptable appearance of the autocorrelations of the log returns. On the other hand, the autocorrelations for the absolute and squared returns remains positive without a desirable more pronounced decay.

VI. The Red Queen principle and Co-evolution in CHASM

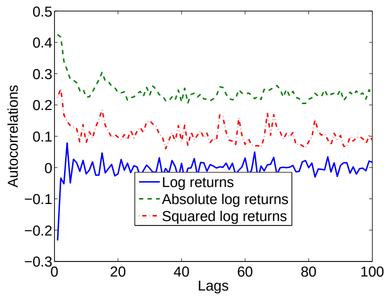
Co-evolution is said to take place when two or more lineages have an impact on each other's selection mechanism and cause



(a)



(b)



(c)

Fig. 17. Autocorrelations for different lags of the log returns, absolute log returns and squared log returns on the Base Case with trading proportion of 30%, 90% and 100%.

changes on each other that increases fitness. This mutual adaptation happens whenever certain ecological interaction derives into fitness effects for all the participants.

Co-evolution in Evolutionary Computation and in particular in GP, is an important area of study in Computer Science and has been used in the modeling of independent agents in the past [39]. Moreover, the notion of evolving competitively populations and in particular decision trees has been intensively studied in [2], [24] and [25].

We aim to model traders in real markets. Therefore, we consider co-evolution to be of central importance in our work, because in real life the traders certainly have an influence in each other's trading strategies. Additionally, it is common for them to change their strategy if they are not performing well in relation to the other market participants. Therefore, in our opinion it is necessary to include in our market model the co-evolution of the agents' trading strategies. In our market the co-evolution of such strategies will be modeled by using two different mechanisms:

- The endogenous generation of the risky asset's price.
- The regeneration of the technical trader's "strategies". In CHASM, this can be done in a fixed or endogenous way.

In our market, the population of investment rules of each trader co-evolve through the price. The effectiveness of each rule (its forecasting precision) determines the probability of it being selected to be part of the population of the next generation during the evolutionary process. The precision of a certain rule belonging to a particular trader depends on the rules of the other traders because they will have to change their rules if their performance on the market is not good. Therefore, it affects the trader's fitness.

Regarding the learning process, in CHASM the user can model the agents, so that they are able to adapt to the new conditions of the market, either periodically (with a user defined periodicity) or with a periodicity that is endogenously generated by a behavioral constraint known as the Red Queen constraint. CHASM allows those two different conditions to trigger learning because we want to contrast both ways, although we believe that learning in an endogenous way is more realistic.

Adaptation of the market participants with fixed periodicity has been used on previous works, eg. [3]. However, we consider it unrealistic because in real life, the traders do not change their investment strategies (at least not the totality) in strictly fixed way. Moreover, the findings reported in [31] are very revealing about the importance of the learning periodicity. In [31] the authors reported the differences on the price properties due to changes on the frequency of the retraining procedure and the memory length of the agents.

The condition to trigger endogenously the learning process has been modeled in previous works. For example, the notion of self realization was modeled in [8] and [58], which is based on a notion of rank. However, we believe that the traders decision to change their strategy must be motivated by their performance on the market in terms of wealth. It is very likely that this decision is strongly related to the other market participants' performance. This was recognized as an important aspect in artificial financial markets by LeBaron in [28].

To summarize, we believe that the modeling of co-evolution in simulated markets is important. Therefore, we make it a key feature in CHASM. It is possible to model the way in which learning is triggered in two different ways: with fixed periodicity or endogenously. In CHASM we are able to model such forms of adaptation and we will report the results of some experiments later in this section.

A. *Individual versus Social Learning*

In the case that an Evolutionary Computation technique is being used to model the agents' learning process, there are two main possibilities for such modelling: Single Population (SP) or Multiple Population (MP).

When SP is used, each individual of the population represents an economic agent. This means that there is just one evolutionary mechanism driving the simulation and the agents have no control over it. On the other hand, in the case of MP each agent possesses a population of individuals and "runs" an evolutionary mechanism by herself.

There are advantages and disadvantages when using either of the two approaches. Nevertheless, we can point out to the ones that we believe are the most significant: when a SP evolutionary process is being used, the individual intelligence is not being explicitly modeled, when a MP evolutionary process is being used, social learning must be represented in an explicit way.

In [8] and in a later and extended work [58] the authors explore two different architectures of an Artificial Stock Market. In the first one, they propose basically a SP-GP mechanism enhanced with another SP-GP mechanism known as the "Business School". In the second one, the authors propose a MP-GP mechanism for the agents and in addition they preserve the "Business School" as well. In the Santa Fe's experiments, they use a MP-GA and the change of behavior is exogenously imposed by the experimenters [3].

Our approach is similar to the architecture in [58] in the sense that each of the agents run a GP mechanism. However, in our market there is no "Business School" or any other explicit mechanism of social interaction. In our case, the interaction through the price (and some other elements) is enough to replicate the stylized facts. Besides, we believe that the inclusion of "trend following" indicators as part of the agents information, can replace an explicit model of interaction among them. Moreover, as it was stated at the beginning of this work the modeling or implications of social interactions in financial markets is beyond the scope of our research.

B. The Red Queen in CHASM

In our research we have a market populated by a co-evolving population of agents, each attempting to enhance its fitness relative to others. This is inspired by the Red Queen principle, based on the observation made to Alice by the Red Queen in Lewis Caroll's *Through the Looking Glass:* "in this place it takes all the running you can do, to keep in the same place".

The Red Queen principle was originally proposed by the evolutionary biologist Leigh van Valen in [57] as a metaphor of a co-evolutionary arms race between species. In cases in which the competition for scarce resources rules the behavior of the participants; the important performance measure is relative to the other individuals involved in such arms race.

The Red Queen principle has been studied in Computer Science, more specifically in competitive co-evolution [46], [44]. In particular, in [9] the authors propose some means to measure the progress in computer simulated co-evolution. The Red Queen effect has been also studied in the context of Economics in the past. In [48], the author claims that the evolution of intelligence itself is hypothesized to arise as a Red Queen type arms race giving rise to Machiavellian behavior in social interactions. In [49] the author describes the relation between the evolution of complex organisms, the reasons behind sexual reproduction, the emergence of high intelligence and the Red Queen effect.

In competitive co-evolution, the Red Queen principle, therefore, entails constraints on performance enhancement of all individuals, if each is to maintain *status quo* in relative fitness measured by an index relating to aggregate performance.

In CHASM, the Red Queen principle will be modeled through the Red Queen Constraint. Such behavioural constraint will force a trader to search for new investment rules whenever she is being "left behind". More specifically, a trader will launch a GP mechanism considering the most recent information (the most recent price history and the relevant indicators) whenever her wealth is below the population's average wealth. As it was recognized in [28] by LeBaron: *"A trader's performance depends critically on the behavior of others"*.

C. Experimental design

To experimentally test the impact of the Red Queen principle, we designed a set of experiments in which the agents will retrain in a fixed or endogenous way. Then, we observe the differences in the statistical properties of the stock returns and the agents' wealth distribution.

In such experiments we defined some study cases which we considered to be interesting. This was taking into account the experience gained with the experiments described in the previous section. Such study cases were different variations of the factors that we identified as important in the previous phase of experimentation.

We first executed the simulation program with the parameters that we consider to be appropriate to get realistic statistical behavior of the log returns. Afterwards, we observed the price behavior and we performed a series of statistical tests to identify if the price presented the stylized facts. In case the stylized facts were not replicated we tried another parameter constellation and so on until we got the desired properties. After getting the appropriate parameters, we executed the market with the same setting but allowing the agents to learn with fixed periodicity. Then, we reported the statistics of such execution. For the experiments reported in this paper learning with fixed periodicity was triggered every one thousand trading periods.

Using the same configuration of the two previous experiments, we allowed the agents to retrain in an endogenous way, this means that we turned on the Red Queen Constraint. Then, we observed the price and the statistical properties of the logarithmic returns. For the experiments reported in this paper, the Red Queen constraint was implemented by retraining the agents whenever their wealth was below the total population's average wealth.

The vertical scales of the different graphs may differ and make comparisons more difficult. However, the change of scale is due to the different behaviour of the price and log returns; therefore, to apply the same scale to all the graphs might cause difficulty to appreciate the behaviour of the price or log returns. The case of the differences on the horizontal scales is due to the computational cost of the market's execution when the Red Queen constraint is activated. The cases where we report the price and log returns under Red Queen retraining will normally have less data points than the cases where such learning does not happen.

D. Case studies and results

Exploring the model as described in the previous section, we developed a series of experiments by changing key features and parameters of the simulation, already identified as important. We have different base cases in which the statistical properties of the price will be reported for different parameters. After tuning the simulation under each case of study to obtain interesting prices and statistics, the simulation is executed again with learning taking place. This allowed us to observe the changes on the statistical properties of price due to the different types of learning.

We will present four out of the possible eight combinations of three important features (Computing Power, Information, and Time Horizon and desired Rate of Return) that we detected as important in the previous phase of experimentation. The reason for the exclusion of the other four cases depends on the specific case. The excluded cases are the following:

- Heterogeneous Computing Power, Heterogeneous Information and Heterogeneous Time Horizon and Return.
- Heterogeneous Computing Power, Heterogeneous Information and Homogeneous Time Horizon and Return.
- Heterogeneous Computing Power, Homogeneous Information and Homogeneous Time Horizon and Return.
- Homogeneous Computing Power, Homogeneous Information and Homogeneous Time Horizon and Return.

The reason for the exclusion of the first three cases is that our previous results showed that heterogeneity on the computational capabilities proved to be counterproductive in our quest for stylized facts. In all the experiments that we performed with heterogeneous computational capabilities, the results were similar in the sense that the price did not resemble by any means prices in real financial markets.

The reason for the exclusion of the last case is that we detected that complete homogeneity was not good neither in our aim. The results of the last case (not reported here) showed a monotonous price (always increasing or always decreasing), ie, there was a sort of consensus and a self fulfilling behavior of the price. We believe that this was mainly caused by the homogeneity of agents on these important features that we detected.

1) Statistics: For each of the experiments' log returns we will report: basic descriptive statistics, the result of the Jacque-Bera Test, the GARCH and ARCH coefficients, the skewness and kurtosis, the correlation coefficient, and the Hill estimator for the 0.1%, 0.5%, 1%, 2.5%, 5%, 10% and 15% most extreme log returns of the experiment's respective data series.

The basic descriptive statistics reported here are the mean, median, minimum, maximum and the standard deviation. Well known stylized facts on stock market returns are that on a weekly, daily and higher frequencies they fail to be normally distributed and they are also unpredictable. By the Jacque-Bera test, we find significant departures from normality for the returns for all the runs. Another stylized fact related to normality is excess kurtosis, the kurtosis for a normal distribution should be equal to 3. However, kurtosis in financial time series is commonly larger that 3.

The correlation coefficient reported here is the autocorrelation with one day difference. Nevertheless, we will report as well the autocorrelation for several different lags. Additionally, we will report the autocorrelation of the absolute log returns and the squared log returns. The autocorrelation of the absolute and squared log returns will allow us to investigate the phenomena known as volatility clustering. Empirical studies in various stock indexes and stock prices have shown that the autocorrelation function of the squared returns remains positive and decays slowly over several days. The autocorrelation function can be defined in the following way:

$$C(\tau) = corr(r(t, \delta t), r(t + \tau, \delta t)) \tag{17}$$

2) Case 1: Heterogeneous Computing Power, Homogeneous Information and Heterogeneous Time Horizon and Return.: In this study case we have seven groups of technical traders with heterogeneous computing capabilities, homogeneous information, heterogeneous desired return and time horizon.

The purpose of testing and analyzing this specific case was to verify the impact that the asymmetry in computational power (in terms of the GP mechanism) has in the market. In other words, heterogeneity in computing power refers to two of the GP parameters assigned to the agents: population size and number of generations. We identified these two parameters as the most relevant ones for measuring the computational capabilities of the agents. In this experiment, the first group of agents was provided with the smallest number of generations and population size (10 and 50 respectively), the second group had a bigger number of generations and population size than the first group (50 and 100 respectively), the third had bigger numbers in both parameters than the second group (75 and 250 respectively), the forth was the most competent of all the different groups (100 generations and 500 population size) and then we started to reduce the population size and number of generations for the remaining three groups.

The obtained prices were the least successful to replicate the stylized facts in comparison to the other cases. We expected such results to a certain extent because it is very important that the behavior of the agents is closer to reality, we need competent traders to participate in our market. Otherwise, the price and its statistical properties do not resemble real data. We found very unrealistic behavior of the price even with learning taking place, in fact the price behavior is worst when learning happens as we can see in Figure 18.

Figures 18(a) and 18(b) show the price and log returns without learning taking place. The next two figures 18(c) and 18(d) show the price and log returns when learning with fixed periodicity takes place. Finally, the figures 18(e) and 18(f) show the price and log returns when the Red Queen constraint is applied. In such figure, we can observe that the price, when learning is not taking place, looks somehow realistic. However,

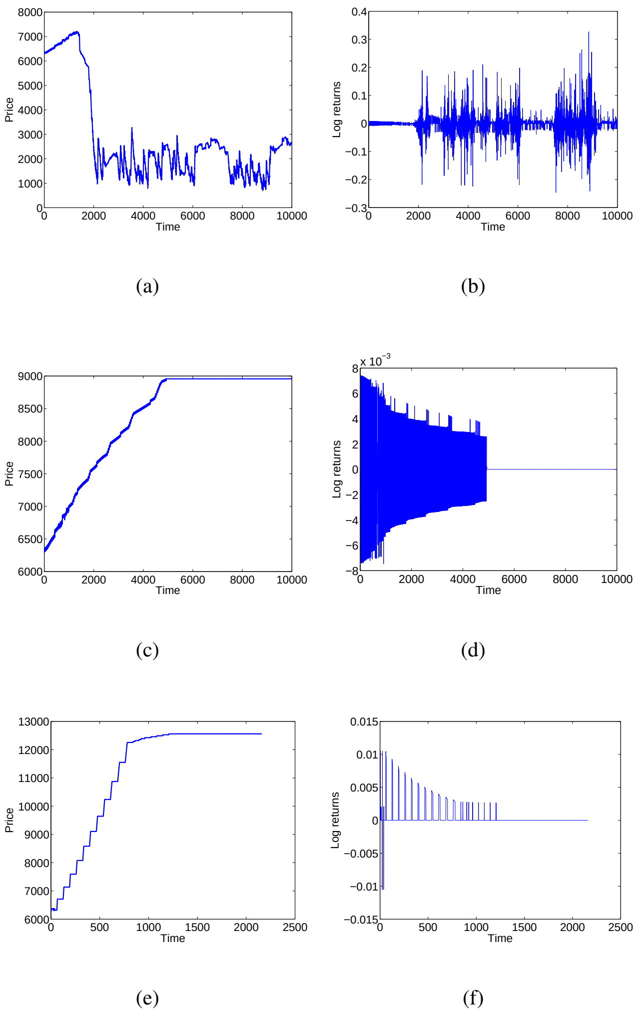


Fig. 18. Case 1 prices and log returns

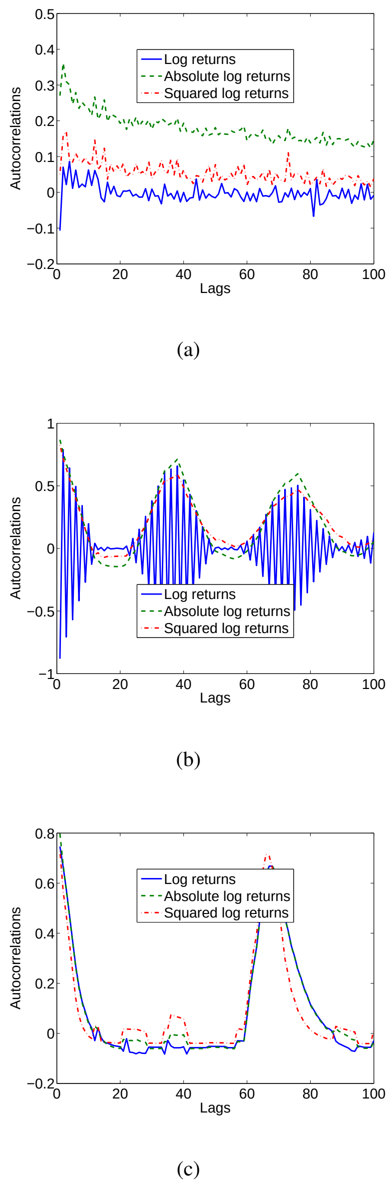


Fig. 19. Autocorrelations for different lags of the log returns, absolute log returns and squared log returns on Case 1 with no learning taking place, learning with fixed periodicity and learning under the Red Queen constraint.

if we observe the other two cases (when learning takes place); then, it was clear that under this conditions and parameter constellation we would not get any closer to the stylized facts.

In Figures 18(c) and 18(e) we can appreciate the sort of things that can go wrong in artificial financial markets research and why is difficult to find the conditions under which the simulated market resembles real markets. By looking at such figures we can see that the agents' perception about the price converged after a certain period and then there was no trading activity. In other words, there were no bids or offers and the agents' were essentially maintaining their current position. The trading activity stopped around the trading period 5000 in the case where learning was taking place in a fixed way. In the case where learning was controlled by the Red Queen constraint, the same thing happened much earlier (around the trading period 1250).

Figure 19 shows the autocorrelations for different lags of Case 1 (No learning, Fixed learning and learning with the Red Queen constraint). Regarding the autocorrelations of the above mentioned three cases, we can see that the case where learning does not take place is the more realistic. On the other hand, in the cases where learning takes place we observe a very unrealistic behaviour of the autocorrelations for the log returns, the absolute and the squared log returns.

Finally, Table III shows the statistics corresponding to this first case. In addition to the standard descriptive statistics we are reporting the correlation, GARCH and ARCH coefficients, skewness, kurtosis, the Hill estimator for different tail sizes and the Jacque Bera test for normality. In such table we can see that the values reported are not very well related to the stylized facts that we are looking for, despite that normality is rejected. In particular the values for the Hill estimator for the fixed learning case are well out of the desired range.

3) Case 2: Homogeneous Computing Power, Heterogeneous Information and Homogeneous Time Horizon and Return.:

TABLE III
STATISTICS FOR THE LOG RETURNS CASE 1

Statistics	**No Learning**	**Learning Fixed**	**Red Queen**
Mean	-0.000091	0.0000353	0.00032
Median	0	0	0
Minimum	-0.261032	-0.007471	-0.010513
Maximum	0.399649	0.007448	0.010535
Std. Dev.	0.023713	0.001721	0.001389
GARCH coefficient	0.916433	0.355824	0.965349
ARCH coefficient	0.083566	0.644174	0.034649
Skewness	0.982907	0.100449	3.79522
Kurtosis	38.6423	9.30956	26.2017
J-B Test H value	1	1	1
Corr. coefficient	-0.108092	-0.881107	0.746318
AlphaHill 0.1 %	3.6429	221.124	
AlphaHill 0.5 %	2.99165	42.2454	
AlphaHill 1 %	2.97188	13.2267	5.69426
AlphaHill 2.5 %	1.88164	3.88784	2.97623
AlphaHill 5 %	1.45753	3.12363	2.06187
AlphaHill 10 %	1.13026	0.451349	
AlphaHill 15 %	0.715213		

In this case we have seven groups of technical traders with homogeneous computing capabilities, heterogeneous information, homogeneous desired return and time horizon. This configuration is the same as for the base case reported on Subsection V-B.

The purpose on the setting up of this case was to study the role that the information, modeled here by the different indicators, has on the price formation and wealth distribution. For that purpose, the computational capabilities of the agents will be homogeneous and the same will happen with the desired return and time horizon. Therefore, the only difference between the trader groups is the information set they will use to create the investment rules. For example, the agents in the first group will use just the moving average indicators to create decision rules, the second group of agents will use the trading breaking rules to do the same, the third group will use filter rules, the forth group will use the volatility, the fifth will use a momentum indicator, the sixth a moving average indicator based on the momentum and the last group will use all the indicators.

Figures 20(a) and 20(b) show the price and log returns without learning taking place. The next two figures 20(c) and 20(d) show the price and log returns when learning with fixed periodicity takes place. Finally, figures 20(e) and 20(f) show the price and log returns when the Red Queen constraint is applied.

In this case, we can see that having traders with more computing power creates more interesting price dynamics. Additionally, we can observe the impact that the heterogeneity in the use of information has on the price dynamics. It is perceivable from Figure 20 that more realistic price dynamics emerged in comparison with Case 1 and that the log returns present volatility clustering in the three different cases.

Figure 21 shows the autocorrelation for different lags of Case 2. In such figure, we can see that the autocorrelations for the experiments where learning does not take place, behave in a realistic way to a certain extent (in particular the autocorrelations of the log returns). When learning in a fixed way takes place, the autocorrelations behave less reasonable, with the autocorrelation of the squared returns being practically zero for all lags. The case where the Red Queen constraint is applied is the one that reports the best behaviour for all the autocorrelations. We observed in the Red Queen constraint experiment that there is a positive autocorrelation of the absolute and squared log returns for short lags, even the decay for such case is quite acceptable.

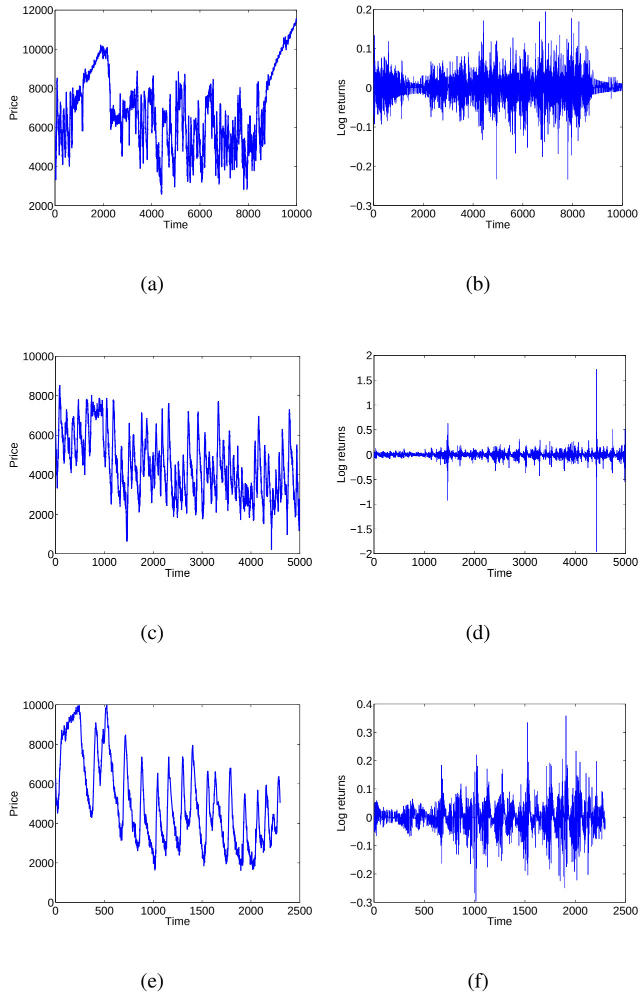
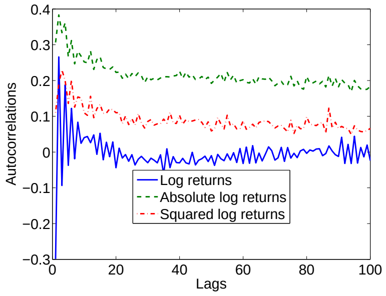


Fig. 20. Case 2 prices and log returns

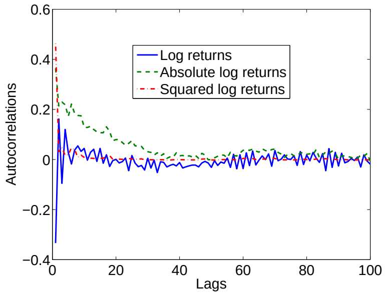
Table IV shows the statistics corresponding to the second of the analyzed cases. We can observe from the statistics, as it was already revealed by the figures, that this case is very good in reproducing the stylized facts. This suggest that heterogeneity in the information used is a key factor to reproduce our desirable statistical properties of the log returns.

4) Case 3: Homogeneous Computing Power, Homogeneous Information and Heterogeneous Time Horizon and Return.: In this case we have seven groups of technical traders with homogeneous computing capabilities, homogeneous information and heterogeneous desired return and time horizon.

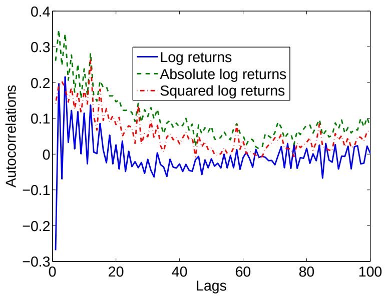
The purpose of this experiment was to verify the importance of the heterogeneity in the agents' time horizon and desired



(a)



(b)



(c)

Fig. 21. Autocorrelations for different lags of the log returns, absolute log returns and squared log returns on Case 2 with no learning taking place, learning with fixed periodicity and learning under the Red Queen constraint.

rate of return in reproducing the statistical properties in real financial markets.

In this experiment, for the first group of agents, we chose a small time horizon and a desired rate of return that made sense for such time horizon (otherwise we would have experienced the same problems that we had at the beginning of our research). For the second group we chose a slightly bigger time horizon and again a desired rate of return that was reasonable to expect in such time. We increased the time horizon and chose the respective desired rate of return for the remaining five groups. With this organization, we had the first groups with small time horizon and rate of return and the last groups with the parameters taking bigger values.

Figures 22(a) and 22(b) show the price and log returns

TABLE IV
STATISTICS FOR THE LOG RETURNS CASE 2

Statistics	**No Learning**	**Learning Fixed**	**Red Queen**
Mean	0.000122	-0.000091	-0.000097
Median	0	0	0
Minimum	-0.129137	-0.552234	-0.296821
Maximum	0.151981	0.516802	0.358635
Std. Dev.	0.013666	0.060209	0.051374
GARCH coefficient	0.991996	0.845768	0.870907
ARCH coefficient	0.007227	0.15423	0.129091
Skewness	0.148694	-0.305522	0.030099
Kurtosis	19.5229	14.4732	8.14092
J-B Test H value	1	1	1
Corr. coefficient	-0.307564	-0.268753	-0.268612
AlphaHill 0.1 %	4.72968	3.173	
AlphaHill 0.5 %	4.0303	2.92985	
AlphaHill 1 %	3.25591	3.04049	4.55034
AlphaHill 2.5 %	2.47495	3.02408	2.94514
AlphaHill 5 %	1.82119	2.64413	2.62655
AlphaHill 10 %	0.88278	2.1084	2.23069
AlphaHill 15 %	0.486301	1.81222	1.90355

without learning taking place. The next two figures 22(c) and 22(d) show the price and log returns when learning with fixed periodicity takes place. Finally the figures 22(e) and 22(f) show the price and log returns when the Red Queen constraint is applied.

We can observe that the obtained dynamics without learning are interesting and realistic. However, when learning in a fixed way takes place, after certain time, there is a sort of agreement and the volatility of the log return starts to decrease. When the Red Queen Constraint is applied, the price exhibits volatility clustering as well and a decrease on it as in the fixed learning case.

Figure 23 shows the autocorrelation for different lags of Case 3. In such figure we can observe well behaved autocorrelations for the three different cases: when there is no learning, when there is learning in fixed time periods and when there is learning under the Red Queen constraint. In particular, we can observe that when learning takes place under the Red Queen constraint, there is a positive autocorrelation for short lags and there is a decay quite similar to the one observed on the two examples in SubsectionI-A. The behaviour of such autocorrelations is the best so far and the Red Queen constraint seems to have an impact on them.

Table V shows the statistics for the case number three. Despite the initial success under this scenario (the values for the reported statistics are within the desired range), the scenario when the Red Queen Constraint is applied is less successful. This would imply that heterogeneity in the agents' time horizon and desired rate of return certainly helps to reproduce stylized facts. However, such heterogeneity is not enough and some more ingredients might be needed to finally obtain the desired statistical properties.

5) Case 4: Homogeneous Computing Power, Heterogeneous Information and Heterogeneous Time Horizon and Return.: In this case we have seven groups of technical traders with homogeneous computing capabilities, heterogeneous information and heterogeneous desired return and time horizon.

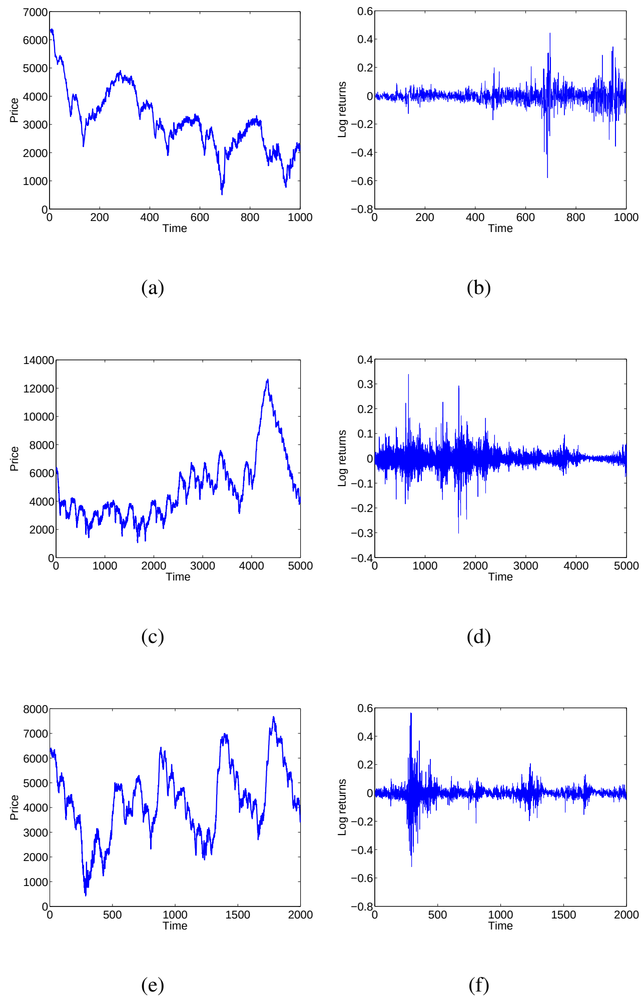


Fig. 22. Case 3 prices and log returns

TABLE V
STATISTICS FOR THE LOG RETURNS CASE 3

Statistics	No Learning	Learning Fixed	Red Queen
Mean	-0.000084	-0.000094	-0.000313
Median	0	0	0
Minimum	-0.227218	-0.302405	-0.52115
Maximum	0.388358	0.338526	0.565525
Std. Dev.	0.020352	0.033338	0.060565
GARCH coefficient	0.933505	0.918561	0.866955
ARCH coefficient	0.066493	0.081437	0.133043
Skewness	0.711244	0.281638	0.241175
Kurtosis	31.6569	14.4124	26.2376
J-B Test H value	1	1	1
Corr. coefficient	-0.245651	-0.391441	-0.392911
AlphaHill 0.1 %	3.32575	5.30302	
AlphaHill 0.5 %	3.19726	3.28499	
AlphaHill 1 %	2.6011	3.21093	2.37963
AlphaHill 2.5 %	2.49727	2.57308	2.19594
AlphaHill 5 %	2.21421	2.36902	1.77109
AlphaHill 10 %	1.86745	2.11074	1.6755
AlphaHill 15 %	1.6483	1.85418	1.60089

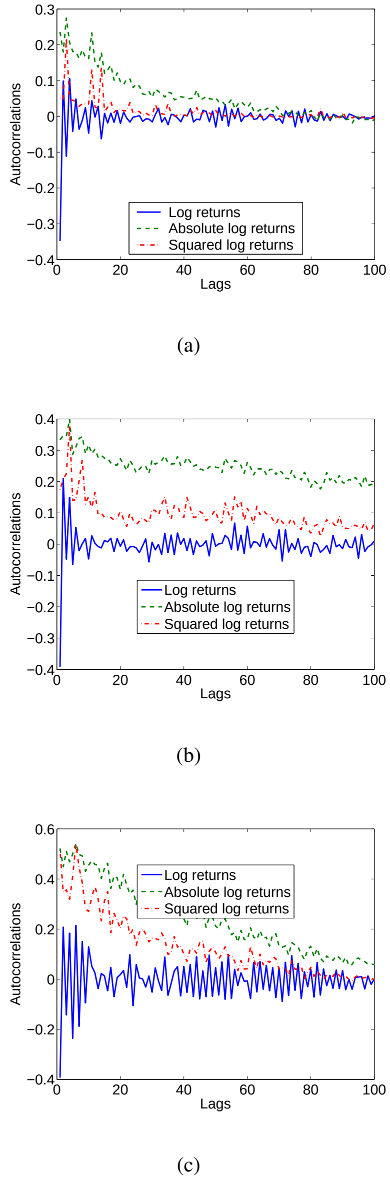
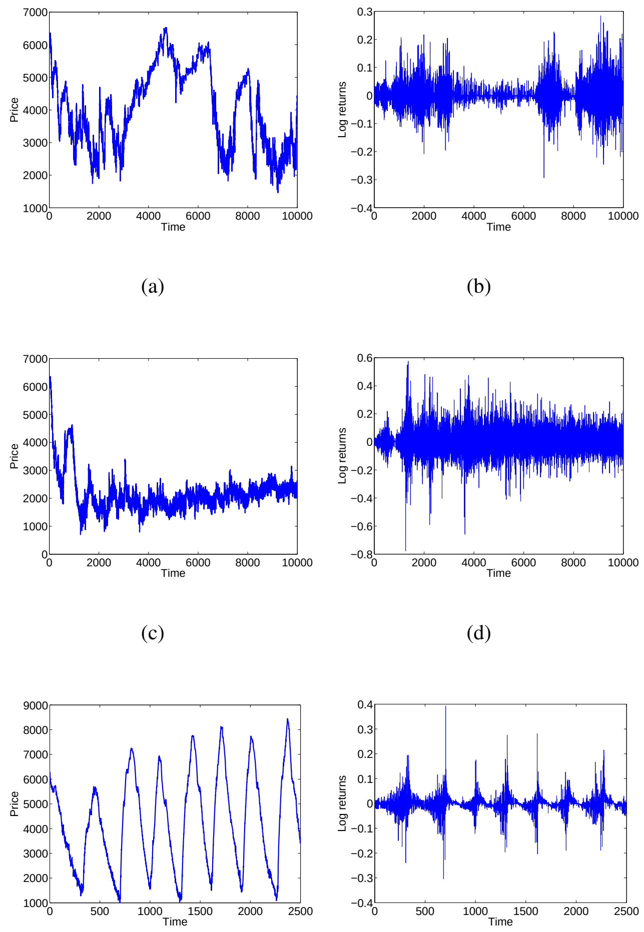


Fig. 23. Autocorrelations for different lags of the log returns, absolute log returns and squared log returns on Case 3 with no learning taking place, learning with fixed periodicity and learning under the Red Queen constraint.

The purpose of this experiment is to explore the impact on the price of the heterogeneity in information, desired return and time horizon. This case would allow us to clarify the importance of the heterogeneity in our market. We expected the price dynamics in this case to be the best of all of our basic cases. Benefitting from the experience of the previous cases, we decided not to make the traders different in terms of the GP mechanism (computational capability). The sources of heterogeneity were: the indicators, the desired rate of return and the time horizon. We assigned the indicators to the different groups in the same way that was described in Case 2. In the case of the time horizon and desired rate of return we proceeded in the same way that was described in the study Case 3.

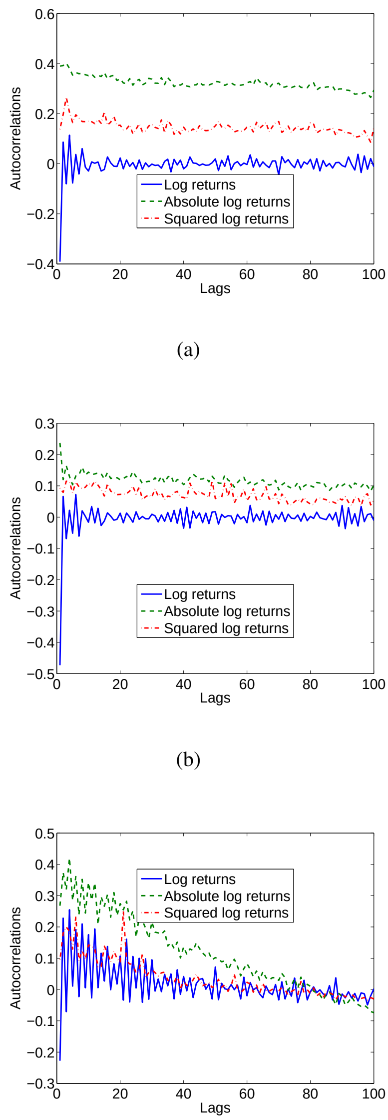


(a) (b)

(c) (d)

(e) (f)

Fig. 24. Case 4 prices and log returns



(a)

(b)

(c)

Fig. 25. Autocorrelations for different lags of the log returns, absolute log returns and squared log returns on Case 4 with no learning taking place, learning with fixed periodicity and learning under the Red Queen constraint.

Figures 24(a) and 24(b) show the price and log returns without learning taking place. The next two figures 24(c) and 24(d) show the price and log returns when learning with fixed periodicity takes place. Finally Figures 24(e) and 24(f) show the price and log returns when the Red Queen constraint is applied.

In this case we can observe a very realistic price being generated without learning taking place, and more importantly: without any exogenous process being used. Additionally, we can observe the emergence of bubbles and crashes during the execution of the experiment in which the Red Queen Constraint is activated (still getting stylized facts), this result is very important since we can clearly appreciate some of the phenomena difficult to explain with current models. It is the case, in our experience, that technical trading might be the responsible for some of the dynamics present in real financial markets. Despite the strong cyclical behaviour of the price, the third experiment of this case is quite revealing on the basis that the price still presents some of the stylized facts.

Figure 25 shows the autocorrelation for different lags of Case 4. The autocorrelations when learning does not take place, behave on a similar way to the most of the cases when learning does happen. In Figure 25(a), we observe a well behaved autocorrelations of the log returns. On the other hand, the autocorrelations for the absolute and squared log returns remains positive for several lags with a small decay. When learning on fixed periods happens, the autocorrelations behave on a similar way to the case when learning does not happen. Nevertheless, the have smaller values for the different lags. The case when the Red Queen constraint is applied, shows similar decay to the reported cases on the SubsectionI-A for the absolute log returns.

Table VI shows the statistics for Case number four. Despite the market cyclical behaviour of the price, in terms of the statistical properties of the log returns, we can say that we have very reasonable values for each of the three different

TABLE VI
STATISTICS FOR THE LOG RETURNS CASE 4

Statistics	**No Learning**	**Learning Fixed**	**Red Queen**
Mean	-0.000036	-0.000092	0.000079
Median	0	-0.000083	-0.000239
Minimum	-0.293954	-0.777021	-0.303987
Maximum	0.283729	0.575629	0.392999
Std. Dev.	0.089299	0.077507	0.038924
GARCH coefficient	0.953031	0.922381	0.864849
ARCH coefficient	0.046967	0.077617	0.135149
Skewness	0.329118	0.152999	0.438554
Kurtosis	8.62664	6.23266	17.5102
J-B Test H value	1	1	1
Corr. coefficient	-0.391261	-0.473053	-0.236828
AlphaHill 0.1 %	13.7083	11.6683	
AlphaHill 0.5 %	4.87626	4.71139	
AlphaHill 1 %	3.79693	3.94017	3.29714
AlphaHill 2.5 %	2.97087	3.85935	2.61319
AlphaHill 5 %	2.60586	3.48402	2.21265
AlphaHill 10 %	1.98593	2.52688	1.7273
AlphaHill 15 %	1.56148	2.11641	1.4732

experiments. We can say that this study case is the best of the four study cases in terms of the statistical properties observed.

VII. CONCLUSIONS

The most important conclusions that we can make about such experiments will be explained in detail in the following subsections.

6) Initial training: The initial phase of training was important in order to generate agents with realistic behavior. At the beginning of the project we did not put enough emphasis on the desired rate of return for a given time horizon. Given our chosen rate of return and time horizon, the classes on the training data were heavily unbalanced. For example, if we asked an unreasonable high return for an unreasonable short time horizon, the class with the highest frequency was the HOLD one. Obviously, the kind of decision trees (investment rules) that were generated by the evolutionary process were trees whose most likely recommendation was to HOLD. This is a well known problem in machine learning and we had to solve it before going any further on the research.

Given the experience gained on the importance of the initial training, we decided to create heterogeneous groups of agents whose desired rate of return and time horizon caused the training data to be equally split on the three different classes (BUY, SELL and HOLD). For example, if a trader wants to achieve a 3% return, she must have a time horizon of at least 5 days because in real financial data changes by 3% over 5 or fewer days are rare; if she wants to get a 4% return, she must wait at least for 7 days because of the few cases present in real financial time series. Otherwise, we could have the classes on the training data highly unbalanced and that would have biased the evolutionary search.

7) Learning and wealth: Learning must have helped to improve the trader's wealth if we were to study its impact on artificial markets. This was a crucial issue for us. In order to prove the importance of learning, we conducted a series of experiments in which we replicated one trader and allowed her to retrain during the simulation of the market. The rest of the traders remained without changes, i.e. without retraining. Afterwards, we compared the wealth of both the original trader and the replicated one. To be able to generalize from this experiments, we performed this task for different traders in different groups. The results in the majority of the cases proved that learning does help to improve performance on wealth terms.

8) Fitness measure: The fitness function, based on the rate of correctness of the GP mechanism, can be used to drive the agent's learning process. This result might sound obvious but it cannot be assumed to be always true, as the rate of correctness of the agents' GP mechanism (working as a classifier) is not necessarily translated into the agents' wealth. The results mentioned in the previous section can be used to justify this point, because we used the rate of correctness for the learning processes.

9) Heterogeneity: Heterogeneity is important for the properties of the simulated returns. By using CHASM we could model heterogeneity in different forms, including: information, computing capability, desired rate of return and time horizon, fundamental like behavior, generation of limit orders, etc.

The different sources of heterogeneity had different effects on the properties of the price and the returns. The experiments described on the Subsection V-B and VI may give us a clear idea of such effects. For example, heterogeneity in the information used by the agents is useful; on the other hand, heterogeneity in computing power is irrelevant.

10) Learning and returns: The learning mechanism does change the statistical properties of the returns. Moreover, we can observe the implications of programming the traders with fixed and Red Queen retraining on the results of the experiments described in Section VI. We can observe that, despite the fact that the simulated prices behave closely to the real prices, when learning happens the properties of the returns reproduce more closely the stylized facts.

11) The Red Queen Principle: The implementation of the behavioral constraint known in this work as the Red Queen constraint is an important component of our model. There have been previous attempts to model the necessity for adaptation to the new conditions on the environment as seen in [8] and [29]. However, we are convinced that the necessity of adaptation should be defined endogenously. The results reported in SectionVI show the difference on the statistical properties of the returns without learning, with learning in fixed periods and with learning driven by the Red Queen constraint. The results show that learning driven by such constraint has a more beneficial impact on the statistical properties of the log-returns in relation with the other two cases.

REFERENCES

[1] F. Allen and R. Karjalainen, "Using genetic algorithms to find technical trading rules," *Journal of Financial Economics*, vol. 51, pp. 245–271, 1999.
[2] P. Angeline, "Genetic Programming and Emergent Intelligence," in *Advances in Genetic Programming* (K. E. Kinnear, Jr., ed.), ch. 4, pp. 75–98, MIT Press, 1994.
[3] W. B. Arthur, J. H. Holland, B. LeBaron, R. G. Palmer, and P. Talyer, "Asset pricing under endogenous expectations in an artificial stock market," in *The Economy as an Evolving Complex System II* (W. B. Arthur, S. Durlauf, and D. Lane, eds.), Addison-Wesley, 1997.

[4] M. P. Austin, G. Bates, M. A. H. Dempster, V. Leemans, and S. N. Williams, "Adaptive systems for foreign exchange trading," *Quantitative Finance*, vol. 4, pp. 37–45, August 2004.

[5] R. J. Bauer, *Genetic Algorithms and Investment Strategies*. John Wiley & Sons, 1994.

[6] T. Brenner, "Agent learning representation advice in modelling economic learning," In K. Judd and L. Tesfatsion, editors, *Handbook of Computational Economics, Volume 2: Agent-Based Computational Economics,*. Elsevier Science B.V., 2005.

[7] W. Brock, J. Lakonishok, and B. LeBaron, "Simple technical trading rules and the stochastic properties of stock returns," *Journal of Finance*, vol. 47, pp. 1731–1764, December 1992.

[8] S.-H. Chen and C.-H. Yeh, "Evolving traders and the business school with genetic programming: A new architecture of the agent-based artificial stock market," *Journal of Economic Dynamics & Control*, vol. 25, pp. 363–393, March 2001.

[9] D. Cliff and G. F. Miller, "Tracking the Red Queen: Measurements of adaptive progress in co-evolutionary simulations," in *Proceedings of the Third European Conference on Advances in Artificial Life*, (London, UK), pp. 200–218, Springer-Verlag, 1995.

[10] R. Cont, "Empirical properties of asset returns: stylized facts and statistical issues," *Quantitative Finance*, vol. 1, pp. 223–236, February 2001.

[11] R. Cont and J.-P. Bouchaud, "Herd behavior and aggregate fluctuations in financial markets," *Macroeconomic Dynamics*, vol. 4, pp. 170–196, 2000.

[12] M. A. H. Dempster, T. W. Payne, Y. Romahi, and G. W. P. Thompson, "Computational learning techniques for intraday FX trading using popular technical indicators," *IEEE Transactions on Neural Networks*, vol. 12, pp. 744–754, July 2001.

[13] M. A. H. Dempster and V. Leemans, "An automated FX trading system using adaptive reinforcement learning," *Expert Systems with Applications*, vol. 30, pp. 543–552, Apr. 2006. Special Issue on Financial Engineering.

[14] B. Edmonds and S. Moss, "Modelling bounded rationality using evolutionary techniques," in *Evolutionary Computing, AISB Workshop*, pp. 31–42, 1997.

[15] B. Edmonds, "Modelling bounded rationality in agent-based simulations using the evolution of mental models," in *Computational Techniques for Modelling Learning in Economics* (T. Brenner, ed.), pp. 305–332, Kluwer, 1999.

[16] B. Edmonds, "Gossip, sexual recombination and the el farol bar: modelling the emergence of heterogeneity," *Artificial Societies and Social Simulation*, vol. 2, 1999.

[17] J. D. Farmer, "Market force, ecology, and evolution," *Industrial and Corporate Change*, vol. 11, pp. 895–953, November 1998.

[18] A. L. Garcia-Almanza and E. P. Tsang, "The repository method for chance discovery in financial forecasting," in *KES2006 10th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems* (Springer-Verlag, ed.), 2006.

[19] A. L. Garcia-Almanza and E. P. Tsang, "Simplifying decision trees learned by genetic algorithms," in *IEEE World Congress on Computational Intelligence*, 2006.

[20] I. Giardina and J.-P. Bouchaud, "Bubbles, crashes and intermittency in agent based market models," *The European Physical Journal B - Condensed Matter*, vol. 31, pp. 421–437, 2003.

[21] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.

[22] B. M. Hill. "A simple general approach to inference about the tail of a distribution," *Annals of Statistics*, vol. 3, pp.1163–1173, 1975.

[23] P. Jefferies, M. Hart, P. Hui, and N. Johnson, "From market games to real-world markets," *The European Physical Journal B - Condensed Matter and Complex Systems*, vol. 20, pp. 493–501, 2001 April.

[24] H. Juille and J. B. Pollack, "Dynamics of co-evolutionary learning," in *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior: From animals to animats 4* (P. Maes, M. J. Mataric, J.-A. Meyer, J. Pollack, and S. W. Wilson, eds.), (Cape Code, USA), pp. 526–534, MIT Press, 9-13 Sept. 1996.

[25] E. V. Siegel, "Competitively evolving decision trees against fixed training cases for natural language processing," in *Advances in Genetic Programming* (K. E. Kinnear, Jr., ed.), ch. 19, pp. 409–423, MIT Press, 1994.

[26] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, Massachusetts: The MIT Press, 1992.

[27] W. B. Langdon and R. Poli. "Fitness causes bloat," In P. K. Chawdhry, R. Roy, and R. K. Pant, editors, *Soft Computing in Engineering Design and Manufacturing*, pp. 13–22, London, Springer-Verlag, 1997.

[28] B. LeBaron, "A builder's guide to agent based financial markets," *Quantitative Finance*, vol. 1, pp. 254–261, 2001.

[29] B. LeBaron, "Empirical regularities from interacting long- and short-memory investors in an agent-based stock market," in *IEEE Transactions on Evolutionary Computation*, vol. 5 of *5*, October 2001.

[30] B. LeBaron, "Evolution and time horizons in an agent-based stock market," *Macroeconomic Dynamics*, vol. 5, pp. 225–254, 2001.

[31] B. LeBaron, W. B. Arthur, and R. G. Palmer, "Time series properties of an artificial stock market," *Journal of Economic Dynamics & Control*, vol. 23, pp. 1487–1516, September 1999.

[32] B. LeBaron "Agent-based financial markets: Matching stylized facts with style." In D. C. Colander, editor, *Post Walrasian Macroeconomics Beyond the Dynamic Stochastic General Equilibrium Model*. Cambridge University Press, Cambridge, UK, July 2006.

[33] J. Li and E. P. K. Tsang, "Investment decision making using FGP: A case study," in *Proceedings of the Congress on Evolutionary Computation* (P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, eds.), vol. 2, (Mayflower Hotel, Washington D.C., USA), pp. 1253–1259, IEEE Press, 6-9 July 1999.

[34] J. Li and E. P. K. Tsang, "Reducing failures in investment recommendations using genetic programming," in *Computing in Economics and Finance*, (Universitat Pompeu Fabra, Barcelona, Spain), 6-8 July 2000.

[35] R. E. Lucas, "Adaptive behavior and economic theory," in R. M. Hogarth and M. W. Reder, editors, *Rational Choice: The Contrast between Economics and Psychology*, pp. 217–242. University of Chicago Press, 1986.

[36] T. Lux, "Herd behaviour, bubbles and crashes," in *The Economic Journal*, vol. 105, pp. 881–896, July 1995.

[37] T. Lux, "The socio-economic dynamics of speculative markets: Interacting agents, chaos, and the fat tails of return distributions." in *Journal of Economic Behavior and Organization*, vol. 33 pp. 143–165, 1998.

[38] T. Lux, "The limiting extremal behaviour of speculative returns: an analysis of intra-daily data from the frankfurt stock exchange." in *Applied Financial Economics*, vol. 11 pp. 299–315, 2001.

[39] J. R. Koza, "Evolution and co-evolution of computer programs to control independent-acting agents," in *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior, 24-28, September 1990* (J.-A. Meyer and S. W. Wilson, eds.), (Paris, France), pp. 366–375, MIT Press, 1991.

[40] S. Markose, E. Tsang, H. Er, and A. Salhi, "Evolutionary arbitrage for FTSE-100 index options and futures," in *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, (COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea), pp. 275–282, IEEE Press, 27-30 May 2001.

[41] S. Markose, E. P. K. Tsang, and S. Martinez-Jaramillo, "The red queen principle and the emergence of efficient financial markets: an agent based approach," in *8th Workshop on economics and heterogeneous interacting agents (WEHIA)* (P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, eds.), vol. 2, (Kiel, Germany), pp. 1253–1259, Springer, May 2003.

[42] S. Martinez-Jaramillo, E. P. K. Tsang, and S. Markose, "Co-evolution of genetic programming based agents in an artificial stock market," in *12th International Conference on Computing in Economics and Finance*, (Limassol, Cyprus), 22-24 June 2006.

[43] C. J. Neely, P. Weller, and R. Dittmar, "Is technical analysis in the foreign exchange market profitable? a genetic programming approach," *Financial and Quantitative Analysis*, vol. 32, pp. 405–426, 1997.

[44] L. Pagie and P. Hogeweg, "Information integration and Red Queen dynamics in coevolutionary optimization," in *Proceedings of the 2000 Congress on Evolutionary Computation CEC00*, (La Jolla Marriott Hotel La Jolla, California, USA), pp. 1260–1267, IEEE Press, 6-9 2000.

[45] R. G. Palmer, W. B. Arthur, J. H. Holland, B. LeBaron, and P. Tyler, "Artificial economic life: A simple model of a stock market," *Physica D*, vol. 75, pp. 264–274, 1994.

[46] J. Paredis, "Coevolving cellular automata: Be aware of the red queen!," in *Proceedings of the 7th Int. Conference on Genetic Algorithms (ICGA 97),* (T. Baeck, ed.), Morgan Kaufmann Publishers, 1997.

[47] R. Poli. "A simple but theoretically-motivated method to control bloat in genetic programming," In *Proceedings of the 6th European Conference*, pp. 204–217. Springer-Verlag, 2003.

[48] A. J. Robson, "The evolution of rationality and the Red Queen," *Journal of Economic Theory*, vol. 111, pp. 1–22, July 2003.

[49] A. J. Robson, "Complex evolutionary systems and the Red Queen," *Economic Journal*, vol. 115, pp. F211–F224, June 2005.

[50] M. P. Taylor and H. Allen, "The use of technical analysis in the foreign exchange market," *Journal of International Money and Finance*, pp. 304–314, June 1992.

[51] L. Tesfatsion, "Guest editorial agent-based modeling of evolutionary economic systems," *IEEE-EC*, vol. 5, pp. 437–441, October 2001.

[52] E. P. K. Tsang and S. Martinez-Jaramillo, "Computational finance," in *IEEE Computational Intelligence Society Newsletter*, pp. 3–8, IEEE Press, 2004.

[53] E. P. K. Tsang, J. Li, and J. M. Butler, "EDDIE beats the bookies," *Software: Practice and Experience*, vol. 28, pp. 1033–1043, August 1998.

[54] E. P. K. Tsang, J. Li, S. Markose, H. Er, A. Salhi, and G. Iori, "EDDIE in financial decision making," *Journal of Management and Economics*, October 2000.

[55] E. P. K. Tsang, P. Yung, and J. Li. "EDDIE-automation, a decision support tool for financial forecasting," *Decision Support Systems, Special Issue on Data Mining for Financial Decision Making*, vol. 37, pp. 559–565, 2004.

[56] E. P. K. Tsang, S. Markose, and H. Er, "Chance discovery in stock index option and future arbitrage," in *New Mathematics and Natural Computation, World Scientific*, vol. 1 of *3*, pp. 435–447, 2005.

[57] L. van Valen, "A new evolutionary law," *Evolutionary Theory*, vol. 1, pp. 1–30, 1973.

[58] C.-H. Yeh and S.-H. Chen, "Toward an integration of social learning and individual learning in agent-based computational stock markets: The approach based on population genetic programming," in *Sixth International Conference on Computing in Economics (CEF'2000)*, (Barcelona, Catalonia, Spain.), 6-8 July 2000.

Serafin Martinez-Jaramillo received his BSc degree in Actuarial Sciences from the National Autonomous University of Mexico (UNAM), Mexico in 1994 and his MSc in Computer Science degree from the University of Essex in 2002. Currently he is working to obtain a PhD in Computational Finance from the University of Essex. His research interest includes constraint satisfaction, genetic programming, financial markets, agent-based economics and computational finance. From May 1997 to September 2001 he worked as a computer systems analyst at the Mexican Central Bank (Banco de Mexico). From October 2006 until now he has been working as a financial analyst at the Mexican Central Bank.

Edward Tsang has a first degree in Business Administration (Major in Finance) and a PhD in Computer Science. He has broad interest in applied artificial intelligence, in particularly computational finance, heuristic search, constraint satisfaction and scheduling. He is currently a professor in computer science at the University of Essex where he leads the Computational Finance Group and Constraint Satisfaction and Optimization Group. He is also the Deputy Director of the Centre for Computational Finance and Economic Agents (CCFEA), an interdisciplinary centre. He chaired the Technical Committee for Computational Finance under the IEEE Computational Intelligence Society in 2004-2005.