Learning-to-Forecast Experiment. A simulation approach with Genetic Algorithm

Anna Safont Andreu

Simone Alfarano

Bachelor's Degree in Economics, 2017/2018

Abstract

In this work, A Genetic Algorithm (GA) is used to study the behavior in a Learning to Forecast Experiment in which short-term expectations have been elicited. In particular, by using the results from a previous experiment with human subjects, the same market is simulated implementing GA. After the training process, the simulation with GAs is able to produce similar results compared to the experiment with human subjects in markets with both negative and positive feedbacks. In addition, simulations in the long-run, i.e. considering 100 and 1000 periods, show a market convergence to the fundamental price of the market and the stability of the GA agents' predictions. We tested how the simulated price reacts by introducing 3 shocks. Finally, the algorithm is also tested in market shocks. The sudden change in the market conditions shows the capability of the algorithm to rapidly adapt and answer to this changes in order to return to the equilibrium conditions.

Keywords: Heterogeneous Expectations · Experiment · Coordination · Convergence · Learning-to-Forecast Experiment.

JEL: D03, G12, C91

I would like to thank Annarita Colasante for her implication and help during the complete development of the algorithm and the design of the simulations.

Contents

List of figures

Learning-to-Forecast Experiment. A simulation approach with Genetic Algorithm

Anna Safont Andreu

1. Introduction

Experimentation has been a key part in the creation of knowledge since the beginning of civilizations. To observe new phenomena, how it changes or also to confirm theories, experiments are reproduced over and over, sometimes simultaneously, in different parts of the world every time any phenomena is being studied.

In the last centuries, the incredible development of technologies allowed for more sophisticated and complex experimental settings and it led to an important advancement in many scientific fields, from medicine to astronomy. Economics took advantage of this innovative tool.

The final technological revolution has been the creation of computers. And with computers, the development of simulations. As computers became more powerful their use has also increased and now simulations are used in different fields, from simulations of high-risk scenarios used in training (military, police) to laboratory experiments, like microchips that simulate living human cell used in drugs tests (Harvard University, n.d.).

Social sciences and psychology rely heavily on experiments to understand the human behavior in concrete scenarios. Despite being a complex source of knowledge, the human brain, the development of Artificial Intelligence has also created a set of tools that can be used to simulate our behavior, creating an alternative method of investigation.

Right now, every investigation in almost any scientific field can rely on experiments and also simulations. The economic investigation is no exception to the rule. On the one side, the use of human subjects is the most reliable source of information to understand the human behavior. Subjects can be selected based on their personal characteristics and knowledge in certain fields. This control over cannot be reproduced by any machine. However, it also has important disadvantages and the most important is the cost of these experiments. The investigation is expensive and its relatively low rate of success turn it into a hardly attractive investment. R&D always finds the same problem with budget constraints and the threat of closing investigations. In relation to this, any possibility of saving some cash would be welcomed.

On the other side, there are simulations. Unless special hardware is needed (does not happen in economics), the only machine necessary is the computers that are already present in every laboratory or office. The software needed to develop simulations can be bought, developed by the same investigators or in the most expensive alternative developed by computer scientists hired exclusively for this purpose. However, Artificial Intelligences (AIs) developed for economic investigations lack excessive complexity and programming languages already carry an important amount of AI code libraries that can be freely used in this process. Any alternative is usually more economical than paying individually for each experiment. This is not the only advantage, though. Another one it's the code itself. As this experiment will show, simulations carry an important number of parameters and equations that are known and can be modified as many times as needed. The program can also be executed endlessly without another cost than the electric power used. The great disadvantage of simulations is the fact that they are indeed not real. Even if their code is complex and sophisticated it will never be the same that, for example, a human mind. Especially in the field or AI, the capacity of this algorithms to imitate human mental processes in heavily limited due to the lack of information. Since, we do not know yet how our minds work, really work, how is going to be possible to imitate them?

Despite the enormous efforts and importance of Artificial Intelligence development nowadays, the algorithms developed are far from being superintelligences capable of succeeding in what Allan Turing defined as the Imitation Game: make humans believe that they are humans too, (Turing, 1950).

Experiments are expensive and rigid, and simulations cheap and flexible, but not reality. The different characteristics of both procedures make them perfectly compatible. Simulations can be used to further test a scenario given all the characteristics possible. After this, the results obtained can be sure as guidelines to develop less, more concrete experiments to further test what simulations have predicted, to confirm whether simulations where right or wrong. This process will be faster as the same simulation is used in more investigations, so the code would have been adapted to the conditions of the experiments.

Another whole possibility is the idea of use experiments as training of AIs. Experiments can alternate both human subjects and computational agents in order to use teach algorithms on how to optimize their behavior against humans, copying their patterns in the process. The interaction between experiments and simulations can perfectly adapt to the needs of the investigations and offer a whole set of alternative routes to proceed.

This work explains also the interaction between an experiment and an algorithm. The experiment was developed first, and then the simulation was programmed and trained using the results of the experiment, showing that this cooperation was enough training for the algorithm and it was able to imitate the human behavior observed in the experiment.

The Learning-to-Forecast Experiment (LtFE) is an experiment in which the subjects are asked to forecast a series of future prices, in this case only the next market price, during a series of periods. At the end of each period, the new market price is set and this value is used by the subjects as a feedback for future predictions.

In this work, results from LtFE has been reproduced using Genetic Algorithms (GAs), trying to emulate the learning process that humans have already applied in a previous experiment with the same market conditions. These heuristics are based on simple linear rules whose variables are both the previous predicted price and the realized one and a realized trend.

In order to avoid any confusion, the word ‘subject’ will always refer to the human subjects from an experiment, while ‘agent’ will always refer to an algorithm. In this work algorithms are always Genetic Algorithm, each agent one. The remaining vocabulary referred to the GA is also very concrete and it’s explained in section 3.

2. Learning-to-Forecast Experiment

2.1. The market

Each market in this simulation consists of 6 agents that will predict the future market prices during several periods. Agents in this market play the role of professional forecasters, whose predictions will affect define the next market equilibrium. This same equilibrium is used as feedback in future predictions creating a learning process.

The experiment considers two different treatments, with positive and negative feedbacks. Market functions have been extracted from Heemeijer et al. (2009). In the negative feedback, the realized price depends negatively on the difference between the average value of the predictions and the fundamental price, while in the positive feedback this relation is positive. The laws of motion for both feedbacks are described in the following equations:

$$p_t = p_f - \frac{1}{1+r}\left(\bar{p}_t^e - p_f\right) + \delta_t \tag{1}$$

$$p_t = p_f + \frac{1}{1+r}\left(\bar{p}_t^e - p_f\right) + \delta_t \tag{2}$$

Where eq(1) represented the negative feedback and eq(2) the positive. The average of the six predictions is represented by $\bar{p}_t^e = \frac{1}{6}\sum_{i=1}^{6} p_{it}^e$ and δ is an iid distributed shock N(0, 0.25). The rational expectations, i.e. the fundamental value, is equal either to 65 or 70. The variable r = 0.05 in both feedbacks and d is equal to 3.25 for a fundamental price of 65 and 3.5 for a fundamental price of 70. These equilibrium prices are calculated as $\frac{d}{r}$.

The experiments have different results depending on the feedback used in the market. On the one hand, the negative feedback generates a price with the opposite behavior compared to the expectations. If the average of the expectations is below the fundamental price, the realized price will be above. In these markets, convergence is fast after some periods of chaotic and uncorrelated predictions. On the other hand, positive feedback has a completely different behavior. In these markets the realized price stays close to the average predictions, being lower than the fundamental if the predictions also are. This much smaller difference between the predictions and the realized price encourages more conservative forecasts and only after a series of predictions their strategies change when their expectations have already overshot the fundamental price. The process repeats itself several times, converging to the fundamental slowly. Compared to real markets, "The positive feedback system mimics the behavior of financial markets where prices typically raise if investors expect positive changes. Conversely, the negative feedback system describes commodities markets where, due to the delay in the production adjustment, market prices move in the opposite direction with respect to expectations. The aim of our experiment is to investigate the impact of the expectation feedback system in the formation of long-run expectations." (Colasante, et al., 2018).

2.2. Expectations formation

Expectations are a key concept to explain the dynamics of any economy, as most decisions depend on the certain degree of the future conditions. Individuals try to optimize their behavior, and to do so they not only take into consideration the current characteristics of the markets but also the most probable evolution, according to their own information and beliefs. Each person is characterized by a set of knowledge, personality patterns, and other characteristics like risk affinity that influence individual behavior that not always coincides with orthodox rationality. These characteristics affect the way they interact with the world and also the economic markets. The process of expectations formation is also influenced by this subjectivity.

This individuality plays two important roles in the formation of expectations. On the one hand, there are other forces besides rationality that can define a person's behavior in economic scenarios. Full rationality is difficult to find in human behavior. On the other hand, rationality cannot explain this individuality. Rational thinking would create similar choices for different people as logic and rationality are common to all intelligent agents. However, all the remaining elements, like personality, amount of knowledge or beliefs can create completely different behaviors that cannot be defined as right or wrong. For example, having more aversion to risk compared to another person is neither a good trait nor bad. Simply different. The resulting behavioral patterns can create very different reactions to the same event in different agents operating in the same role, in the same market. For example, two forecasters can interpret the sudden increase in the market price in different ways and therefore react in a contrary way. However, none of them would be easily considered the appropriate one.

Expectations are defined by uncertainty rather than facts and because of that uncertainty human subjectivity is more present than in other situations. However, the interaction is more complex than that. Expectations, different for each individual, define the decisions taken and those same decisions will have consequences for the future markets.

In the traditional literature, agents' behavior is taken as perfectly rational. A key assumption in classical economics is that the learning process is not important and agents have perfect information about the market. Based on these assumptions, individuals seek to optimize their choices, knowing that there are more market participants whose behavior is as important as their own, from the beginning and the expected economic conditions will be consistent with the future realized prices. The Rational Expectation (RE), as it is understood in financial markets, is based on the idea that participants' objective is always to optimize their behavior and any wrong beliefs or uncanny ideas will disappear given a certain amount of time when people will behave in a perfectly rational way and the price will converge to the fundamental one.

However, in practice, none of these assumptions stand and agents will need to learn about the market before being able to show some rational conduct. According to the literature, empirical predictions rely on two important stylized facts (Colasante, et al., 2018). The first one says that agents usually rely on simple strategies in order to design their future behavior. The second one implies that, even though these heuristics may be simple, the aggregate dynamics can be complex. People modify their current heuristics based on their past performance, but these modifications depend on each agent. The weight of past expectations, past prices or certain events can be completely different from one person to another, creating heterogenous agent behavior. The aggregate outcome is the result of those chosen strategies of all the agents in the market. This continuous feedback defines the learning to forecast process that subjects follow and it allows explaining macroeconomic phenomena through rather simple interactions at the microeconomic level.

Experiments with human subjects allow analyzing this learning process and its inherent heterogeneity and subjective heuristics. However, experiments like this can only be reproduced at a small scale compared with real markets and with much less frequency than the study requires. These great limitations are a reason for considering alternative tools in order to study markets. Is in this moment when computational simulations can offer an alternative way.

3. Genetic Algorithm

In the recent decades, the field of Artificial Intelligence has developed a vast number of algorithms and methods that try to identify the way human brain process data and takes decisions, since the moment the term was founded in Dartmouth College during a summer workshop in 1956. Evolutionary Algorithms are a branch of Artificial Intelligence that optimizes solutions applying the same methods that nature uses with living creatures along generations: adaptation to the environment define the changes of a creature to survive and reach adulthood. Those adults that survived have some changes to reproduce and give birth to a new generation that will inherit that genetic material that created adapted individuals in the first place. This process repeats itself during several periods until the population is highly adapted to the environment and no longer depends on adapting any further. These rough ideas taken from Darwin's Evolutionary Theory can be applied to repeated choices applied in a concrete situation: choices are tested, modified and even combined until an optimal response is created. At that moment the subject taking the choices will stick to one definite response that will be the best one it has been capable of creating.

The Genetic Algorithm is the most popular type of Evolutionary Algorithms and the one chosen for the experiment described in this work, since it's application in Learning-to-Forecast experiments has already been tested (Anufriev, et al., 2012). This algorithm is able to simulate the subjectivity and heterogeneity of human behavior and it that can be applied to a market to create a complete simulation. The GA allows creating agents as independent entities whose behavior depends on their own information and the previous market response, where the other agent's conduct is reflected through the market price. This way each algorithm learns individually and the heuristics utilized by the different agents will remain heterogeneous during the entire experiment. The comparison between humans and Genetic Algorithms behaviors will permit to understand if effectively the subjects that participated in the experiment based their Learning-to-Forecast in simple heuristics similar to the ones used by the GA.

Agents in a market can have several possible strategies in order to define their expectations. Choosing the best one requires to be able to evaluate the different possibilities and modify them, even to combine them into a better one. This process can require several repetitions in which the strategies are refined to the point where the person using them consider its accuracy good enough to stick with it.

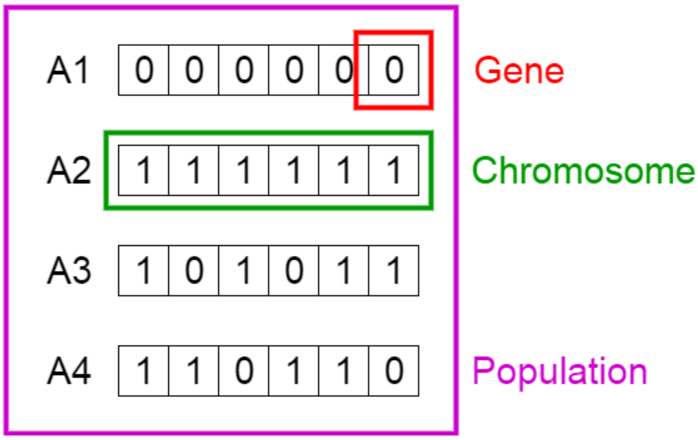
A Genetic Algorithm is a stochastic and metaheuristic method inspired by Darwin's theory of natural evolution in order to solve both constrained and unconstrained optimization problems. Each algorithm considers a given amount of possible solutions and these are modified and mixed in several periods until its solution is considered good enough. These processes of modification and mixing are heavily inspired by the natural laws of evolution.

Even though algorithms using some principles of evolution as a learning process appeared during the 50s and 60s, it was John Holland in the 70s the responsible of its popularization through the formalization of the framework to predict the quality of future generations. However, by that time the limitations of computers made its application almost impossible during a decade. Nowadays, they are a valid and efficient optimization method used in many different fields and also inside more complex Artificial Intelligence systems.

In addition to the bioinspired methods already discussed, the GA works with data in an alternative method, different from the most orthodox idea of numbers. Instead of working with the whole value of a number and applying arithmetic operations to modify its value, potential solutions are considered and codified as chains of data, similar to DNA in living creatures, and it’s manipulated by changing the values of certain positions, ignoring the meaning of those values. They are usually in a binary system, but GAs can also use alternative methods for decimal values.

Since they are based on natural selection, the vocabulary used is the same. The most important terms are:

- Population: the complete set of possible solutions considered by each agent. The strategy used by the algorithm will be selected from this population, the one it considers best. This number is constant along the whole algorithm and specified by the user according to its preferences.
- Chromosome: synonym for the individual. Populations are then sets of chromosomes. It refers specifically to the chain of values the algorithm works with, not the interpretation of that chain in the problem context.
- Gene: each one of the values in the chromosome chain. Binary number or decimal. Its positions are called locus.



GAs deal with different ideas, or strategies, as if they were a population of living creatures. The best ones are more likely to be considered, not only as a solution themselves but also as a source to create new solutions by mixing them up or making small changes through mutations. Individuals are the ideas the GA, the agent, is considering using as strategies. This evolutionary process defines the method an agent thinks and makes choices.

From now on and to avoid any confusion the vocabulary referring to the algorithm will be very concrete. And the agent is a GA and therefore the entity that makes predictions. Each agent considers a set of possible strategies to create those predictions and each one of them is an individual or a chromosome. The whole set of individuals at a given period is the population of each agent. Finally, the chromosome the agent considers best will be its chosen strategy and given the chromosome, the prediction will be calculated.

3.1. Procedure

A GA starts with a series of parameters specified by the user, which are the number of chromosomes in the population and the number of periods (even though some implementation of this algorithm stop when there are too noticeable changes from one population to the next). The concrete methods used in the selection, crossover, mutation, and substitution phases are also specified before the algorithm is executed and remain the same along the execution. After this process the previous population of potential solutions will be modified into a new one, whose adaptation to the problem's context, the market in this experiment, will be better than the previous one.

Selection

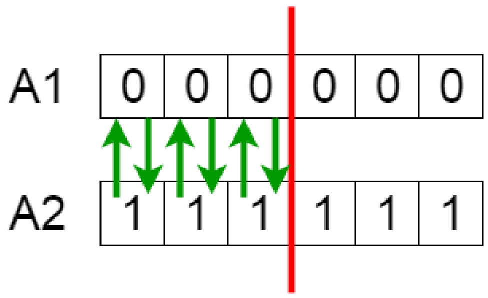
To create a new population of solutions, the first step consists of selecting two previous solutions, known as parents, for each couple of new solutions. Two parents will have two children for the next period.

In order to select the parents, each solution has a score associated. This score is obtained through a fitness function, a mathematical function capable of measuring the suitability of a solution in the context of the problem. Since most of the problems in which Genetic Algorithms are considered consist of optimization problems, this same function is often used as the fitness function. In economics experiments, the fitness function scores how close the prediction is to the market equilibrium, taking into account the market structure. The fitness function used in this simulation is provided in the next subsection.

These scores are used to obtain the probabilities. The probabilities measure the how much each individual contributes to the total score of the population. An individual with a score double than another will have two times more chances to be selected than the one with less score. Individuals can be selected multiple times. A couple of parents can be formed by two copies of the same individual.

Crossover

Once the couples of parents are selected, their information is mixed so their children will combine parts of both parents, without adding further changes. After the creation of one child, the remaining genes will form the other one. Parents not always are mixed, but it depends on a crossover probability, a constant value during all the experiment that is set at the beginning.



A5	1	1	1	0	0	0
A6	0	0	0	1	1	1

There are also many different methods, but with binary chains, they all consider one or more positions where the parent whose information is being copied changes. For example, if a crossover considers a position 5 in chromosomes of length 10, positions 1 to 5 will be a copy of the first parent and then from 6 to 10 of the second parent. The other child will be the opposite: from position 1 to 5 will be a copy of the second parent, and then from 6 to 10 of the first.

When parents are two copies of the same individual, both children will remain the same, exact copies of their only parent.

Mutation

Up to this moment, there are no ways to consider further possibilities, and all solutions are a combination of the previous ones. Since the initial population is initialized with completely random values, the final solution is limited by this random initialization. To avoid this restriction that can limit the power of the Genetic Algorithm, a new step is added.

The mutation phase allows any position of any solution to be changed to a random value. Given a certain probability, the mutation probability applied to all genes in all solutions, the value can be randomly modified. From 0 is switched to 1 and vice versa.

Substitution

Once the new offspring has been mutated, it’s necessary to decide whether the new individuals will be used to construct the next population or discarded. Populations can keep a certain amount of solutions from one generation to the next or create a whole new population. It depends on the chosen method.

Another important concept in GAs is elitism. The concept of elitism refers to the phenomenon in which the best solution in a population, according to the fitness score, always survives and becomes a member of the next generation.

This ends the algorithm procedure. The number of times it's repeated can be chosen by the investigator or established by the stability of the populations.

3.2. Specific settings for this simulation

The entire code of this experiment has been programmed by me. The code includes the Genetic Algorithm itself, the market structure and all the data structures needed to coordinate several algorithms in the same simulation. The Genetic Algorithm is based on the example of Jacobson, (Jacobson & Kanber, 2015). The programming language used is Java 10 (JDK 10.0.1) and the IDE Eclipse Oxygen.

Data codification

In this experiment, the solutions have been previously codified in binary. Two values, α, and β, integrate each chromosome and are binary strings of length 20. The decoding from binary to decimal is obtained with the following equation:

$$\theta_d = A + (B - A) \sum_{l=1}^{20} \frac{g_{l-1} 2^{l-1}}{2^{20} - 1} \tag{3}$$

The parameter A refers to the starting value of the interval and B to the end, of either α and β. This process is applied to both strings, which have been programmed as an independent. The parameter g represents the gene, the value of each position that will be either 0 or 1. Positions work from right to left and the gap of value 1 refers to the fact that positions in data structures in computer science always start with position 0. Since $2^0 = 1$, it will be possible to codify odd numbers. The summation represents how binary numbers are transformed to decimal by a sum of numbers 2 whose exponential represents the position of the binary numbers and multiplied each one by the binary digit. For example:

$$1101_b = 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 8 + 4 + 0 + 1 = 13$$

Given the limitation of a binary chain to represent real values there is going to be a certain loss of precision. However, this could be solved by using longer chains to codify the data and this change does not affect to the algorithm execution efficiency. The length in this experiment will remain constant at 20 in concordance with the previous experiment with GAs in economics, (Anufriev, et al., 2012).

Equations

The Genetic Algorithm uses most of the parameters, functions, and methods of the GA described in Anufriev et al. (2012). Some modifications have been applied in order to compare the results with an experiment with human subjects realized in the Laboratory of Experimental Economics of this university.

The price prediction of this algorithm does not work by predicting the following price itself, but by establishing the values of the parameters of a function. This function considers the weight, represented as α, given both last realized price and last own prediction. The expectation is calculated as a weighted average of both. A trend is also added, based on the difference between this last realized price and the previous one. The parameter β measures the weight assigned to that trend, even if it's going to follow the opposite direction compared to the previous two market prices when it's value is negative.

$$p_t^e = \alpha p_{t-1} + (1-\alpha)p_{t-1}^e + \beta(p_{t-1} - p_{t-2}) \tag{4}$$

Variables p_{t-1} and p_{t-2} represent the last two realized prices, while p_{t-1}^e is the last expectation created by the agent itself. The value of α falls in the interval [0, 1] and β uses [-1, 1], so this trend can be added or subtracted to the expectation created.

The expected price itself does not appear within the algorithm. Once the process of selection, crossover, mutation, and substitution is completed each agent selects its best choice, based on the fitness score. At this moment the value of the predictions is translated, according to this equation, and all agents' predictions define the new market price. Further explanations appear in the next section.

Another function extracted from Anufriev et al. (2012) is the fitness equation. As it has been explained, this function is used in the selection process to define the probabilities of an individual to be selected as a future parent. Before the substitution too, the fitness of the new offspring is calculated in order to compare their adaptation to their parents.

$$f_t = exp(-\gamma(p_t^e - p_t)^2) \tag{5}$$

The fitness is basically a function of the difference between the expectation that the current individual would create (based on previous expectation and market prices) and the previous market price. The fundamental price does not appear, so the fitness only checks how close the individual is to create a new expectation similar to the previous price. In this function, the parameter γ represents the sensibility applied to the divergence between the expectation and the realized market price. After experimenting with different values, the final version of the experiment considered this value constant at 1.

Probabilities

An important concept in GAs are probabilities, that play a key role in the behavior of the algorithm. While selection and substitution phases are always present, neither crossover nor mutation occur always. This values again are taken from Anufriev et al. (2012).

$P_{crossover}$ = probability of a selected couple of parents to crossover. If not, the same individuals are considered the offspring. The value used is 0.9 for each couple of parents.

$P_{mutation}$ = probability of an individual gene to mutate. The value used is 0.01 for each gene in each individual.

These values optimize the behavior of the algorithm in the experimental context. The proper performance of genetic algorithms with binary chains uses similar probabilities (Palma Méndez & Marín Morales, 2008).

Methods

The selection method is 'fitness proportionate selection'. In this method, the percentage of fitness of each individual, compared to the total fitness accumulated in the total population, defines the probability of each individual to be selected.

The crossover method is 'simple crossover'. For each couple selected for crossover, a random number is selected to be the crossover position. Previous positions will be from one parent and the posterior ones from the other.

The mutation method is ‘simple mutation’. The mutation probability is applied to every gene from each individual in the population. If chosen, the value of the gene will be switched from 0 to 1 and vice versa.

The substitution method is original from the economics field. Since the Genetic Algorithm is imitating a reasoning process from a human subject, its considered that only those strategies with better results than the previous ones will survive to the next period. Based on this idea, the only children that will be selected are the ones whose fitness is higher than the parents. If fail, the parents are the ones that will be selected and the children disregarded.

Once the substitution phase is applied, the best individual from each algorithm will be considered the chosen one to form the expectations. With the expectations, obtained from each agent, the new market price will be calculated and then a new period will begin.

Differences with original genetic algorithms

This application of a Genetic Algorithm alters two fundamental principles of this algorithm. The original concept only considers the population of the last period and ignores all the rest. The process followed to get the best solution possible is not important not checked. Because of this, during the substitution phase, it is usually allowed to introduce new individuals that apparently have worse scores since they are also a source of new chain fragments. The fitness of the offspring is originally never considered during this phase, only during the selection.

The algorithm always defines the number of individuals that will change from one population to the next one. The algorithm creates the exact number of new individuals needed, so they are always introduced in the new population. However, the old population, if it is only partially substituted, will need to decide which individuals will survive. Methods can vary a lot, but the most common ones select the best individuals to survive to the next population. Another common method is selecting the survivors randomly.

Parallelism with human behavior

The application of a Genetic Algorithm, in a learning processes framework, equate the learning process followed by a human agent with the one followed by the algorithm itself during its execution. Both processes start with more or less random strategies and finalize with a concrete strategy, considered the best one by the creator, human or machine. Because of this, a single genetic algorithm will represent the whole learning process of a subject. During each step, the algorithm generates a new population that also represents a new set of strategies used by the human subject, and the next strategy used will be selected from this set (or population). Each one of these steps is an opportunity to refine the subject's (or agent's) strategies.

5. The simulation

The experiment itself consisted in repeating the same LtFE done with human subjects and compare the results between both, subjects and agents, in order to understand whether the algorithm is able to replicate human choices or not.

However, the use of Artificial Intelligence shows some limitations. A key concept in any algorithm is how the information about the environment in which the algorithm operates is represented. What it understands and how and what ignores. This representation is always simplistic and very limited compared to the way human subjects understand it, so, whenever a decision made by humans escape these limited framework algorithms tend to fail. To avoid crashes or any sort of unexpected failure, this lack of knowledge is usually replaced by randomness.

In this experiment with GAs, the lack of information appears, as it has already been explained, in the initial period. The algorithm initializes the first population randomly, while humans do not act that arbitrarily. In fact, human subjects tend to choose the initial price close to the middle of the interval of values this price can have. In the experiment, prices can fluctuate from 0 to 100, so the initial prices will always be close to 50. The algorithm, using random values for the parameters and also random values for all the previous prices and expectations, will show higher variability. Because of this, it has been necessary to create a gap of two periods (the same to periods that used random prices and expectations in the fitness) to synchronize the experiment and algorithm.

5.1. The database

Once the algorithm is completely programmed, it's necessary to teach him how to behave like a human. After its development, data from the experiment with human subjects have been added in order to train it to simulate the behavior of human subjects. The experimental database comes from the experiment published in Colasante et al. (2018). That session took place in the Laboratory of Experimental Economics at University Jaume I, I which 90 subjects participated in 15 different market simulations for both negative and positive feedbacks. In this experiment, the predictions created by the subjects not only reach a short-run prediction for the next period but also a series of long-run predictions. The GA is only capable of reproducing the short-run predictions.

This experimental data is responsible for teaching the algorithm how humans behave. In order to train the algorithm is necessary to introduce some these experimental results and use it during the algorithm execution. Other algorithms in AI include some sort of training process, but not Genetic Algorithms. Because of that part of the feedback used to evaluate future choices (in the fitness equation) was extracted from the experiment instead of the algorithm results themselves during the first periods of the simulation. The data used as training was the market prices created with the one-step-ahead subject predictions. These prices were created with the expectations of the subjects, the same way the algorithm does it. All fifteen markets data have been included and the choice of the market used to train the algorithm is completely random. To do this training, 20 periods of prices have been extracted from the experiment and used in the fitness function instead of the ones generated by the algorithm.

5.2. Results

Comparison with the experimental results (short-run)

In each period, the algorithm chooses the individual in its population with the highest fitness value, so in each period six heuristics, one per agent, are used. In order to study the evolution of these choices, it's convergence and coordination have been computed along with the experimental ones and used for further analysis of the algorithm behavior.

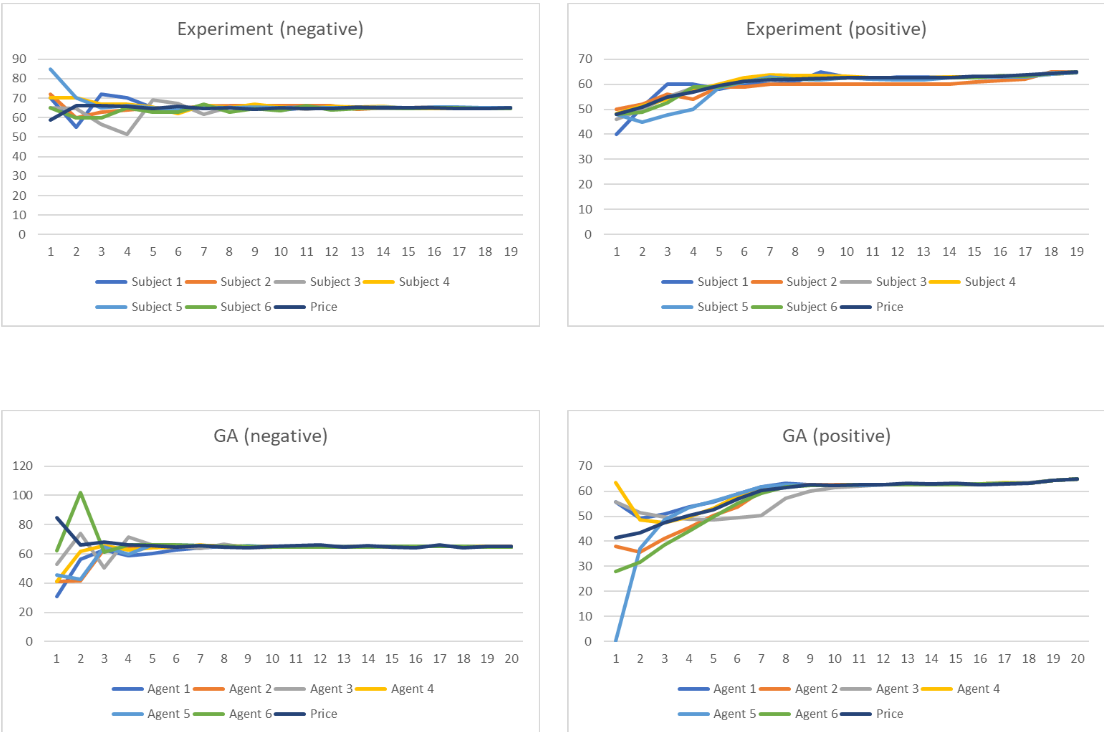


Figure 1. Examples of simulation of twenty periods for negative and positive feedback systems. The two graphs above correspond to the results of the experiments with human subjects, two randomly selected groups, and the ones below the algorithm executions that used those same groups as training.

The market prices and expectations have different patterns depending on the market feedback system and these characteristics are shared by both the experiment and the algorithm. It has been shown in Colasante et al. (2018) that in the positive feedback market there is a fast coordination of expectations and slow convergence to the fundamental value. The opposite is observed in the negative feedback system.

Convergence of the predictions to the fundamental price and coordination of the agents' expectations are the statistics used to ascertain the similarity between the experiment and the algorithm. Convergence has been calculated as the Relative Mean Square Error (RMSE) of the agent's choices:

$$RMSE_t = \sqrt{\frac{\sum_{i=1}^{6}(p_{it}^{e} - p_f)^2}{6}} \quad (6)$$

The variable p_{it}^{e} represents the expectation of each agent and p_f the fundamental price. The denominator depends on the number of agents.

For each period the value is an average of one hundred algorithm executions. The experimental value is also the average of the one hundred experimental databases randomly selected, one in each algorithm run, providing the market prices.

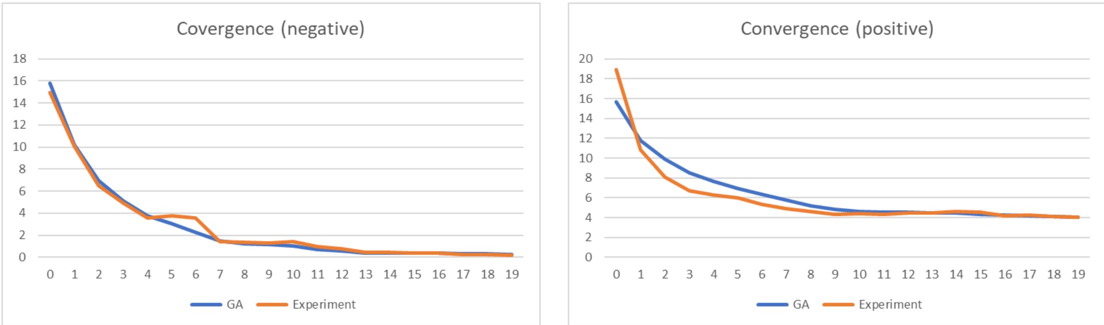


Figure 2. Convergence results for both negative and positive feedbacks. It includes the experiment and the algorithm.

Convergence in the positive feedback the algorithm follows the same pattern than the experiments, softening the shape of the curve. In the positive feedback, the experiment results in higher values during the first periods but decreases faster than the algorithm. In both cases, the second half of the experiment shows steady values that are very similar for the experiment and the algorithm.

An important fact of the positive feedback is that rational expectations don't explain the convergence as well as in the negative one and prices in this feedback does not converge to the fundamental one.

The other statistic considered is the coordination, also calculated as the average of one hundred algorithm runs of the standard deviation of the six agent's choices for each period.

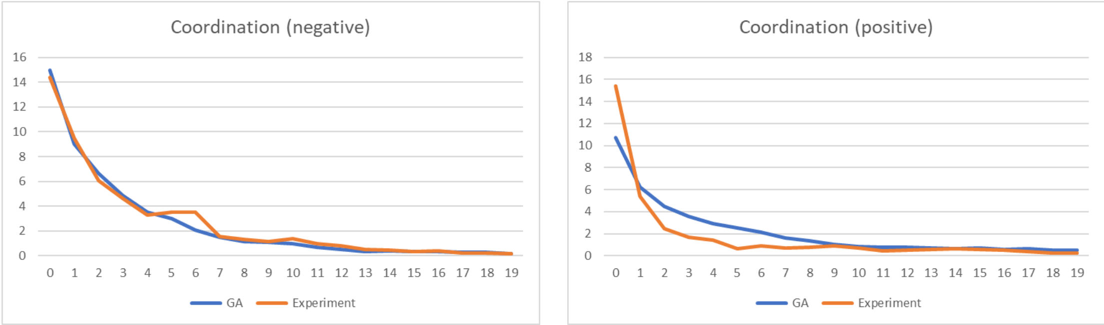


Figure 3. Coordination results for both negative and positive feedbacks. It includes the experiment and the algorithm.

The results are very similar for the coordination comparison. In the positive feedback, the algorithm follows again the same pattern than the experiments but softer. In the positive feedback, the experiment also results in higher values during the first periods but decreases faster than the algorithm. Again, the second half shows similar and steady values for both feedbacks.

As literature predicts, in both feedbacks convergence and coordination follow different speeds. Negative feedback markets should have faster convergence than coordination than convergence and positive ones the opposite, faster coordination than convergence. Experimental results show that positive markets effectively follow the predictions while negative ones show very similar evolutions for both convergence and coordination, however, these results are similar to the ones in the experiment.

Once the GA and the experiment results are similar, the heterogeneity of the heuristics used by the algorithm can be studied in order to understand if effectively the agent's behavior can converge towards the market equilibrium at a macro level, even though the individual expectations heuristics may not converge following the same pattern.

Long-run convergences

After the comparison of twenty periods of algorithms with the experiments, it can be established that effectively the algorithm is capable of reproducing the human behavior.

The number of periods of the algorithm can be increased to further study its asymptotic behavior. A new execution of the algorithm, of both feedbacks, and one thousand periods showed the next results. These results correspond to a single simulation (run) of the algorithm. Again, the experimental session selected to train the algorithm during the first twenty periods is randomly selected.

Negative feedback

For comparison purposes, these are all the results, price convergence, predictions coordination, prediction values, and price, of a negative feedback experiment with human subjects (20 periods):

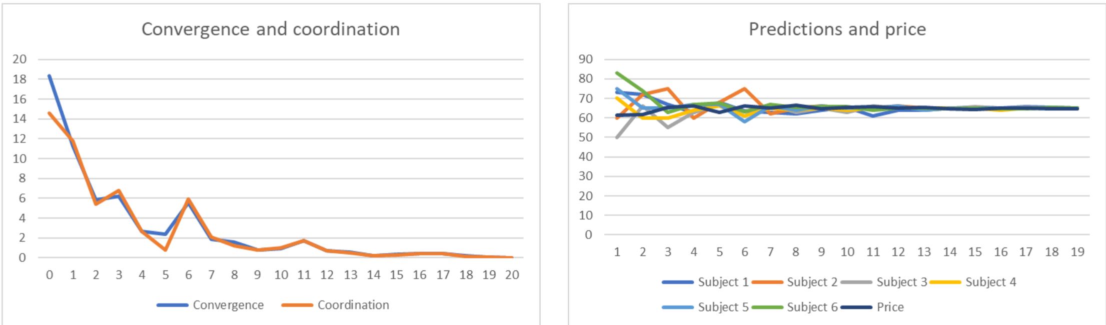


Figure 4. Results of an experimental group with negative feedback.

While the algorithm behavior is this (1000 periods):

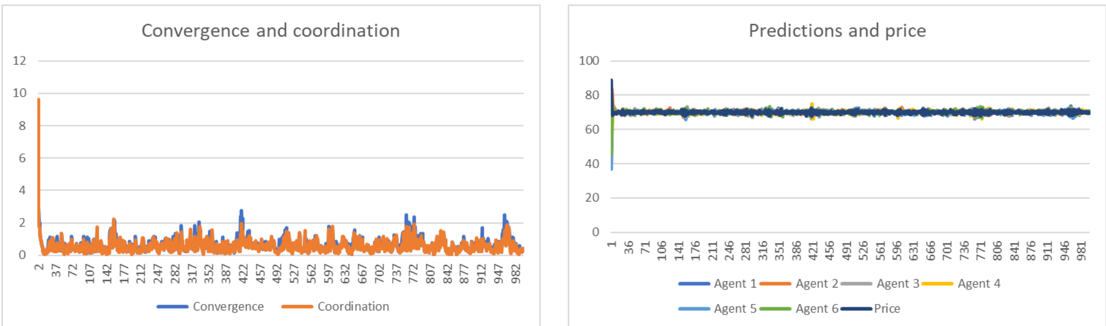


Figure 5. Results of a simulation with negative feedback.

The analysis of the convergence and coordination show a rapid decrease of both statistics, as happened in the twenty periods experiment, followed by periods in which the values remain close to zero, with only five phases in which they reached values higher than two. The learning process is successful after approximately twenty periods and after this remains stable.

Positive feedback

For comparison purposes, these are all the results, price convergence, predictions coordination, prediction values, and price, of a positive experiment with human subjects (20 periods):

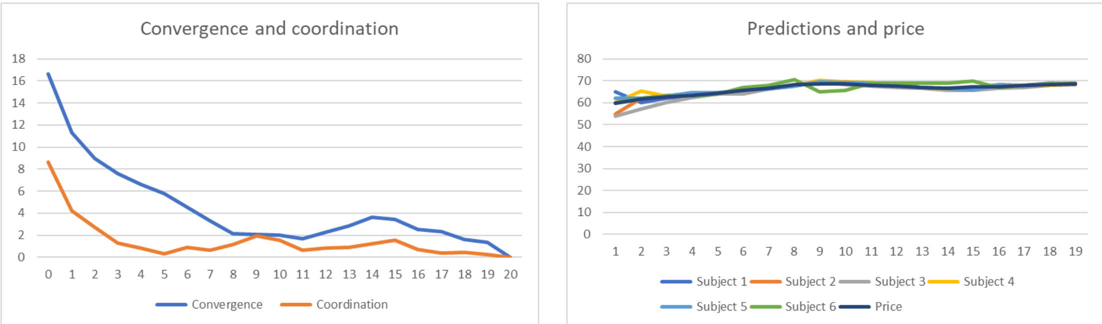


Figure 6. Results of an experimental group with positive feedback.

While the algorithm behavior is this (1000 periods):

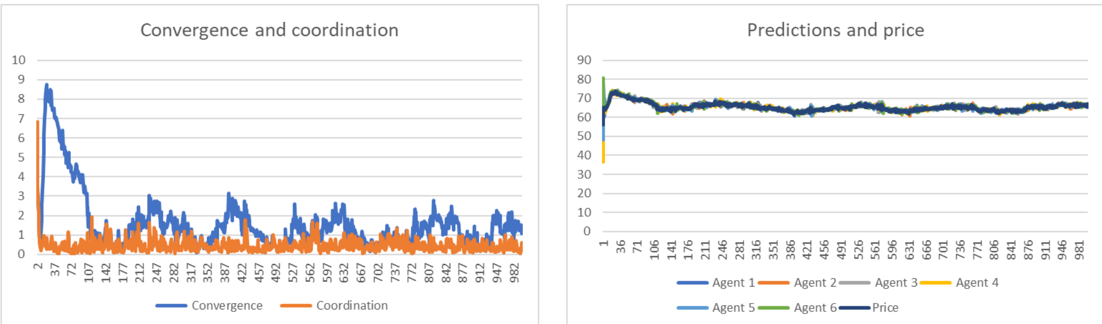


Figure 7. Results of a simulation with positive feedback.

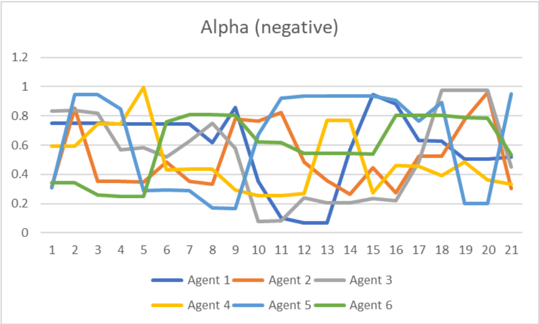
The behavior of the positive feedback simulation shows a different result compared to the one with negative feedback. As the literature predicts, the price converges much slower in the positive feedback because agents need to coordinate much better in order to change the price trend. This price fluctuates around the fundamental price, staying some periods with prices higher than the fundamental followed by periods with values lower than this fundamental and then again higher. The process is slow but there exists convergence. The first graph shows how the coordination of the agents is rapidly obtained and remains along the whole simulation, but even in these circumstances, the convergence of the price is slow and unstable. However, the peaks of this convergence decrease as the experiment continues.

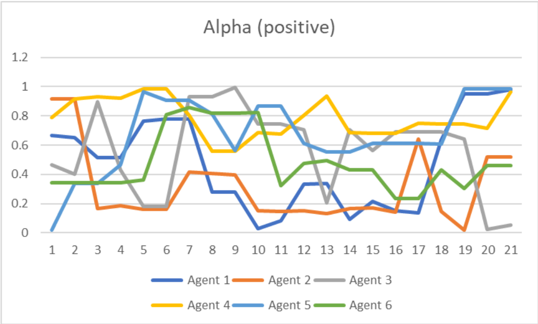
Heterogeneity analysis

Once the aggregate behavior of the algorithm shows indeed that it can replicate the human behavior, the next question is how the strategies of the algorithm evolve during the simulations, more importantly if there is any learning in the use of these strategies and after a certain number of periods both values, α and β become stable around certain values.

To check this, it has been analyzed the evolution of α and β in simulations with a different number of periods.

20 periods





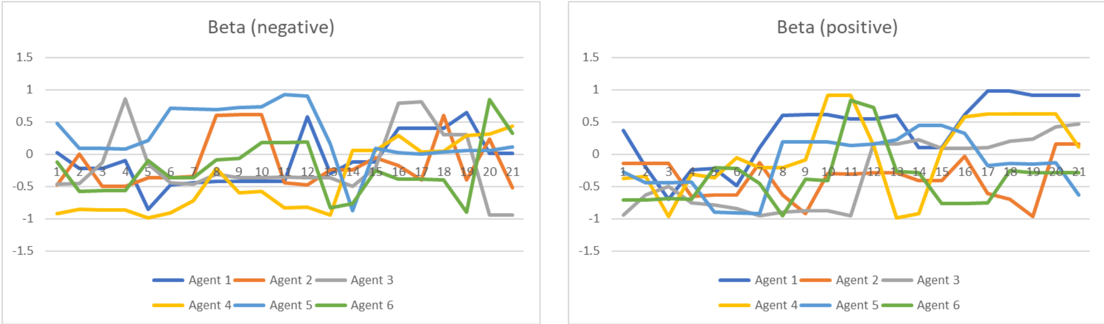
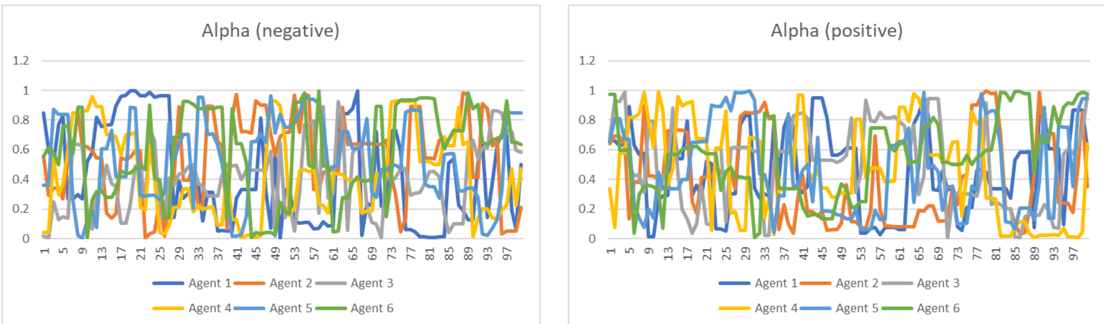


Figure 8. Evolution of the values α and β in a simulation with 20 periods, for both negative and positive feedbacks.

The first simulation considers only twenty periods, in concordance with the length of the experiments with human subjects. As the graphs show these values can change completely from one period to the next for the same agent and this variability lasts the whole simulation. In most periods agents' values are distributed along all the interval [0, 1] for α and [-1, 1] for β.

100 periods

The next simulation considered a number of periods five times bigger than the previous one.



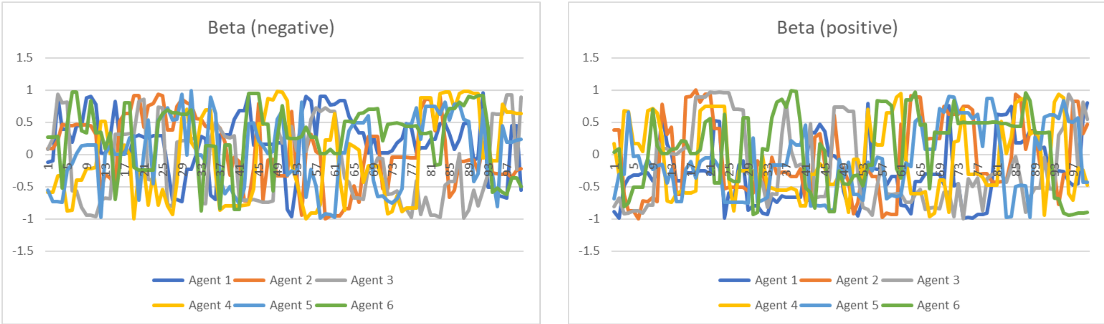


Figure 9. Evolution of the values α and β in a simulation with 20 periods, for both feedbacks.

The results are the same and the agent's choices almost seem to be random. There is no proof to consider any sort of convergence or stability during the complete execution.

1000 periods

Finally, one thousand periods have been chosen since this is the number of periods considered in the study of long-term convergence in the previous section.

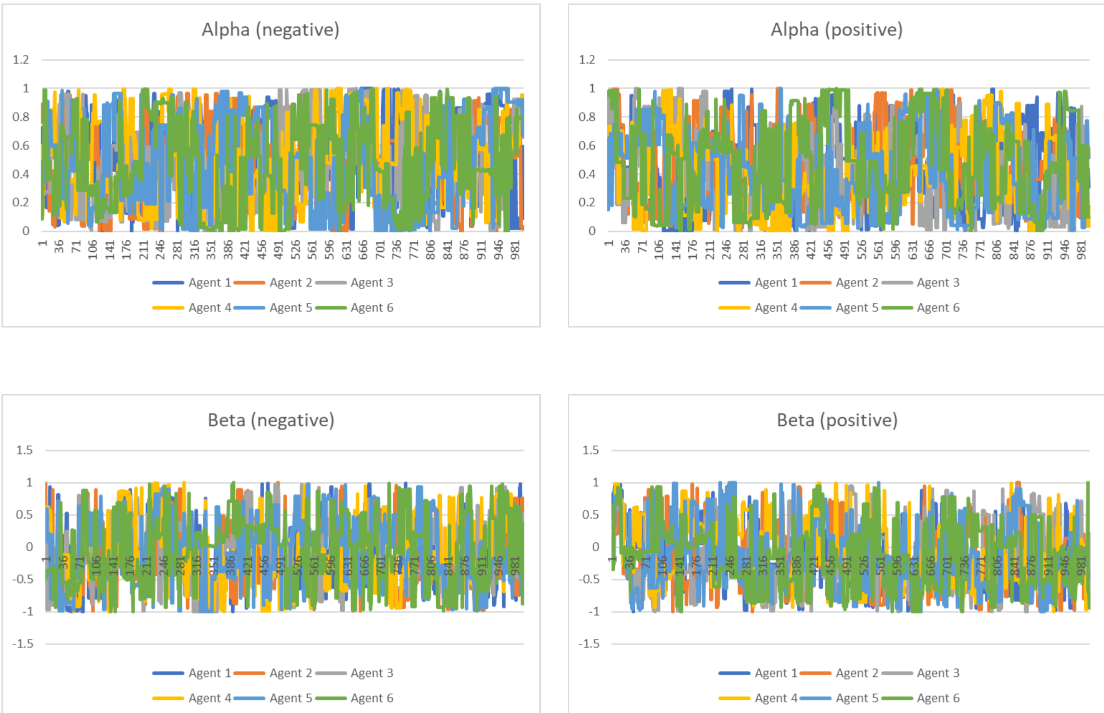


Figure 10. Evolution of the values α and β in a simulation with 20 periods, for both feedbacks.

Again, the result is the same and even though is difficult to appreciate it in graphs with so many periods almost the whole intervals for α and β are colored. This is constant during all periods. These results show how the algorithms do not have any difficulties to change from one strategy to another, given that the all of them will have similar results. The more stable the prices stay the more difficult is for the algorithm to focus on a strategy that can be considered better.

In conclusion, the results seem contradictory. While the market simulations are in concordance with the literature when the strategies used by the algorithm are analyzed the results are illogical. The algorithm is able to simulate the learning process while, in reality, is not learning at all and strategies to predict the future prices are close to being random. The results show how the aggregate behavior, compared to agents individually, can appear contradictory as continuous changes show aggregate convergence.

These results can be understood by checking the expectations formation function. On the one side, α defines the weight the previous price compared to the last prediction. If both values are similar, and they are after the first twenty periods, it does not matter much which one has more weight creating the new prediction. On the other side, β includes a trend that extrapolates the changes from the last two periods to the current expectation. Again, if the difference between two consecutive prices is small, the value of this parameter will not ad substantial changes in the expectation.

An alternative version of the fitness equation

As it has been explained in a previous section, usually GAs are used in optimization problems and the fitness equation chosen is often this function that has to be optimized. However, in this problem, there is no such equation and what the fitness evaluate is how close the prediction is to the last realized market price. So, the structure of the fitness equation is a way to score this proximity but other alternatives could be used to rank with more, or less, sensibility the adjustment of the strategy to the market price. Knowing this, the fitness equation has been modified in order to differentiate more between better chromosomes and worse. To do so a new parameter has been added to the original fitness equation from Anufriev et al. (2012).

$$f_t = exp(-\gamma \boldsymbol{\varepsilon}(p_t^e - p_t)^2) \tag{7}$$

In this new version of the fitness equation, the parameter ε will allow us to modify the fitness sensitivity. Since both behavioral patterns differ, the modifications will be tested separately in both market feedback systems.

Negative feedback

After checking different values for this parameter, ε = 100.000 shows some convergence of the parameters α and β that last hundreds of periods in the second half of the execution. This convergence to steady values could be understood as the end of the learning process, however, the analysis of the convergence of prices and the coordination show a different scenario. As soon as the strategies remain constant, prices move away from the fundamental one rapidly and only when these heuristics return to their variable pattern the convergence returns to normal behaviors.

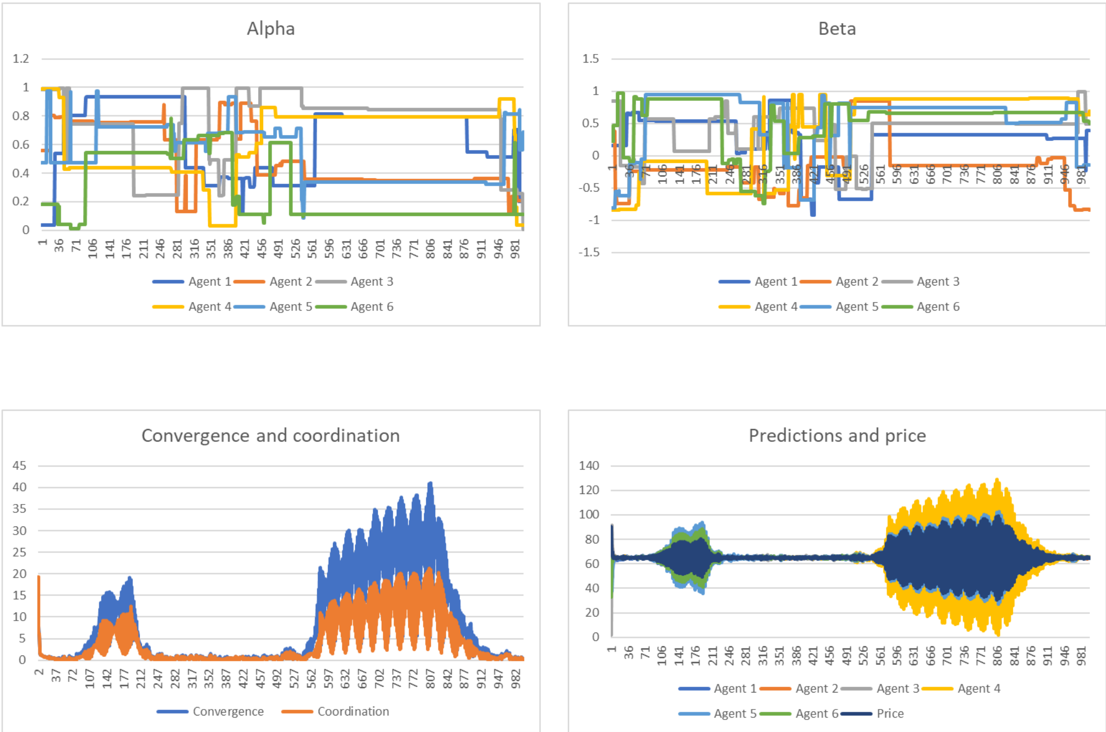


Figure 11. Results of heuristics convergence in negative feedback. Heuristics and market behavior.

In addition, as prices variate so much with steady strategies is impossible for these to remain stable. It will always be a period in which the lack of convergence will be so high that the heuristics will finally change. Even if the behavior of prices is ignored, the values of α and β will not remain constant.

Positive feedback

After checking different values for this parameter, ε = 10.000.000

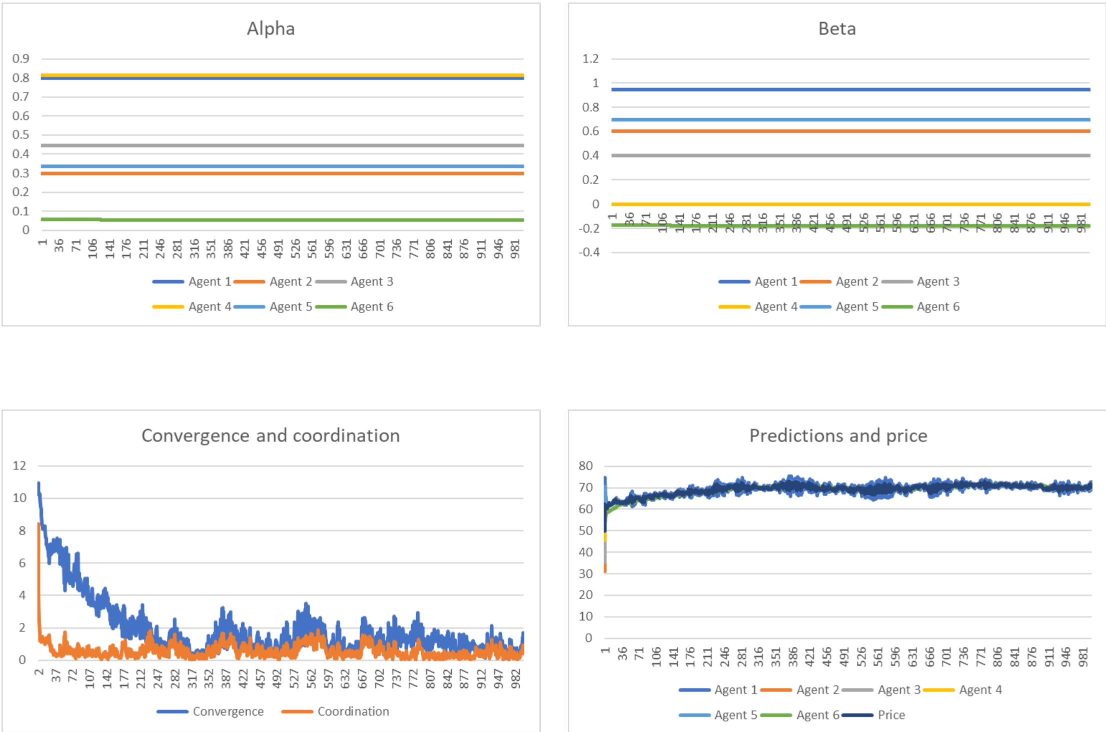


Figure 12. Results of heuristics convergence in negative feedback. Heuristics and market behavior.

With a value of ε = 10.000.000, the analysis of the strategies convergence in positive markets show strategies that remain constant along the whole execution, with punctual small changes. As it can be seen during all the process, this feedback behaves completely different from the negative one. There is not any sort of learning of α and β parameters and the enormous increase in the ε value only achieves a stability of the strategies chosen by each agent since the beginning. This situation could not be considered a failure, as the value of ε does not allow the strategies to change at all. The whole execution is determined by the choices made out of the initial random populations.

However, the result of the converge of prices and coordination give us different information. Even though the heuristics do not change, the market is able to achieve prices similar to the fundamental one. Instead of using around ten periods, like in the version with the original fitness function, two hundred and fifty periods are now needed to achieve convergence values smaller than two. Also, the price shows a variability much higher but the behavior of this strategies compared to the ones in the original are much more similar than the negative feedback algorithm.

In conclusion, the negative market cannot achieve any real convergence of the strategies used by their agents by modifying the fitness equation, adding a parameter that increases the sensibility of this score to better individuals. Despite showing some stability on the values of α and β, once the convergence of prices and the coordination are checked this learning process shows not to be real. The situation in the positive feedback is completely different. By the same method, even though the value of this parameter is heavily increased, the positive feedback markets do not seem to show any convergence, only a strong stability of the strategies used that starts from the beginning of the execution. However, taking these strategies as constant the market is indeed able to reach the fundamental price after many more periods than the original setting though.

The explanation in these two different behaviors can be found in the way each feedback uses the expectations to create the market prices. While the negative feedback generates a price with the opposite behavior compared to the expectations, positive feedback has a completely different behavior and the realized price stays close to the average predictions, being lower than the fundamental if the predictions also are. This much smaller difference between the predictions and the realized price encourages more conservative forecasts and only after a series of predictions their strategies change when their expectations have already overshot the fundamental price. The slow fluctuations around the fundamental price allow prices, expectations and thereby trends to change at smaller intervals than the ones on the negative feedback.

Market shocks

Once the heterogeneity has been analyzed and the limitations of the GA established, they have been tested in a different way.

In real life, markets can show sudden and abrupt changes that are not a direct consequence of the markets themselves but of external factors related to politics, diplomacy or other matters instead. There are many situations that can move a market away from the previous equilibrium and an important field in economic sciences is the study of how the economy responds to these sudden changes to return to the previous equilibrium of reach a new one, depending on the nature of the changes, whether they are structural or not.

GAs allow reproducing this shocks by setting artificially the conditions on the market for a given period of time. Once the algorithm reaches that period the market situation will change completely and in the next periods, the algorithm will try to solve that new situation.

A new simulation has been set, in which the market prices are artificially set for three different periods along the simulation of one hundred periods in total:

Period	**25**	**50**	**75**
Market price set (shock)	95	15	70
Difference from p_f = 65 (negative)	30	50	5
Difference from p_f = 70 (positive)	25	55	0

Depending on the feedback system, the difference between these shocks and the market fundamentals are different.

Negative feedback

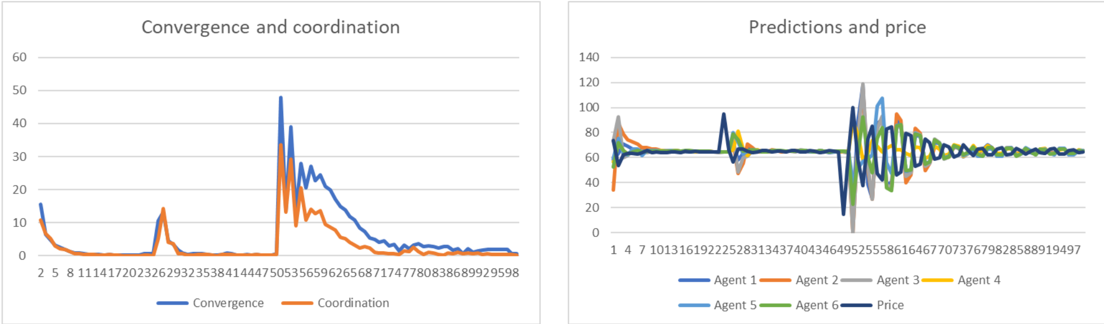


Figure 13. Simulation with three market shocks in periods 25, 50 and 75 in negative feedback.

The response to the first shock, where the market price is set to a value of thirty units higher from the fundamental price is indeed fast and after 5 periods the market is in a situation very similar to the one before the shock. The second one is fifty units higher and even though the difference is almost twice the one in the previous shock, the market response is much bigger. Convergence and coordination are three times bigger than at the beginning of the experiment and the stabilization of this values is slower and more irregular too. Looking at the predictions and price the shock causes a series of market prices that converge to the equilibrium in a chaotic way, again more unstable that the initial periods. The small shock in period seventy-five is completely invisible since the market is already dragging the instability caused by the previous shock.

Positive feedback

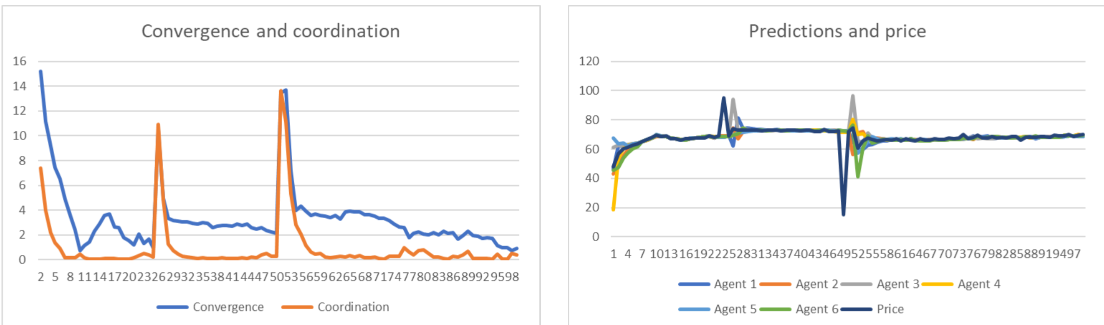


Figure 14. Simulation with three market shocks in periods 25, 50 and 75 in positive feedback.

Positive feedback also shows violent responses to shocks, even though predictions and prices seem more stable, convergence and coordination are slower than in the negative feedback and require many more periods to return to stable values. In fact, convergence is never fast enough to recover completely from one shock before the following one appears. Coordination is faster and only needs ten periods after a shock to return to its normal pattern.

In conclusion, these shocks can be used to study the market answer to certain sudden instabilities that can be much higher than the initial conditions of the simulation. The different ways the algorithm responds to the shocks correspond to the differences between both feedback systems. In the negative one, the persistent fluctuations around the fundamental price are caused by the mechanism that alternates the position of the expectations with the one of the resulting realized price in relation with the fundamental one. The market price jumps from higher to lower values and the difference of this values with the fundamental price is a bit smaller each time. The positive feedback mechanism creates a realized price stays close to the average predictions, being lower than the fundamental if the predictions also are. This much smaller difference between the predictions and the realized price encourages more conservative changes that finally correct the market behavior and reach the equilibrium again faster than the other feedback.

6. Conclusions

In this work, a Genetic Algorithm has been implemented from the scratch and introduced in a program that simulates a market in which the algorithms predict future prices and at the same time influence its value with their predictions. The structure of the algorithm follows the one in Anufriev et al. (2012), taking the predictions and fitness function, and is used to complement the experiment of Colasante et al. (2018), whose experimental data is used in the training process.

After using the experimental market prices in twenty periods experiments it’s shown that effectively the algorithms follow the same convergence and coordination patterns that the experiments so it can be concluded that the training is successful and both the strategies used by the algorithm are complex enough if compared with the ones used by the subjects and also that using experimental market prices in the fitness evaluation is enough training for the GAs.

The second phase included simulation with longer time periods, on thousand, in both feedback systems. The results agree with the literature and show that the negative feedback simulation, once it converges to the ration expectations, remains close to this value with little changes. The analysis of the positive feedback market shows that the convergence process fluctuates around the fundamental price. Even though it is slow, there is convergence.

Once it’s tested that the markets effectively show a convergence towards the fundamental prices, the ration expectations, it is time to study the evolution of the strategies used by the algorithm, and how they change along the execution of the simulation. Three new executions, of twenty, one hundred and also one thousand periods have shown that the heterogeneity of the strategies used by the algorithm is constant and as high as possible, regardless of either the number of periods or the feedback system. Considering this a limitation of the algorithm structure a new approach is considered and, keeping a constant value of one thousand periods, a new parameter has been included in the fitness equation. This parameter accentuates the sensibility of the algorithms to differentiate between better and worse solutions. Testing different values for this parameter is found one in which the negative feedback shows the convergence of strategies for the second half of the experiment. However, this convergence of strategies turns into a chaos of market prices during the same periods. On the other side, the positive feedback does not show any sort of convergence and much higher values for this parameter cause the strategies to remain stable since the beginning of the experiment. Despite this exaggerated stability, the market is capable of reaching the fundamental price after a much longer learning process.

Finally, shocks have been tested. The sudden changes in market prices force the algorithm to readapt and these corrections are shown in the following periods. Each feedback system has a different answer and also a different speed to stabilize again both predictions and price, where positive feedback is more efficient, and convergence and coordination, where the negative feedback is more efficient.

This approach could be improved in other studies with other GAs with more complex methods and equations. Many different implementations could be studied further to find the ones better adapted for this specific experiment framework. Another possibility is to consider alternative AI tools like Artificial Neural Networks or Self-Organizing Maps, which actually are machine learning algorithms contrary to GAs, which are optimization algorithms. However, complex simulations could be designed using more than one method for the different structural modules.

Bibliography

Anufriev, M., Hommes, C. & Makarewicz, T., 2012. [Online] Available at: http://publish.illinois.edu/essummer2013/files/2013/06/paper85.pdf

Colasante, A., Alfarano, S. & Camacho-Cuena, E., 2018. *The term structure of cross-sectional dispersion of expectations in a Learning-to-Forecast Experiment.* [Online] Available at: https://mpra.ub.uni-muenchen.de/84835/

Harvard University, n.d. *Human Organs-on-Chips.* [Online] Available at: https://wyss.harvard.edu/technology/human-organs-on-chips/

Heemeijer, P., Hommes, C., Sonnemans, J. & Tuinstra, J., 2009. Price stability and volatility in markets with positive and negative expectations feedback: An experimental investigation. *Journal of Economic Dynamics & Control 33(5),* pp. 1052-1072.

Jacobson, L. & Kanber, B., 2015. *Genetic Algorithms in Java Basics.* New York: Springer Science + Business Media.

Mallawaarachchi, V., 2017. *Introduction to Genetic Algorithms.* [Online] Available at: https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3

Palma Méndez, J. T. & Marín Morales, R., 2008. *Inteligencia Artificial. Métodos, Técnicas y Aplicaciones.* Madrid: McGraw-Hill.

Russell, S. & Norvig, P., 2010. *Artificial Intelligence. A Modern Approach.* New Jersey: Prentice Hall.

Turing, A. M., 1950. Computing machinery and intelligence. *Mind,* pp. 433-460.

Anna Safont Andreu