

Learning by Genetic Algorithms in Economic Environments

Jasmina Arifovic

SFI WORKING PAPER: 1990--001

SFI Working Papers contain accounts of scientific work of the author(s) and do not necessarily represent the views of the Santa Fe Institute. We accept papers intended for publication in peer-reviewed journals or proceedings volumes, but not papers that have already appeared in print. Except for papers by our external faculty, papers must be based on work done at SFI, inspired by an invited visit to or collaboration at SFI, or funded by an SFI grant.

©NOTICE: This working paper is included by permission of the contributing author(s) as a means to ensure timely distribution of the scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the author(s). It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may be reposted only with the explicit permission of the copyright holder.

www.santafe.edu



SANTA FE INSTITUTE

**LEARNING BY GENETIC ALGORITHMS
IN ECONOMIC ENVIRONMENTS**

Jasmina Arifovic

University of Chicago, Hoover Institution,
and the Santa Fe Institute

October, 1989



Abstract

This paper studies models in which economic agents use genetic algorithms to learn about their environment and to learn about their objective functions.

The main results of the paper, obtained through the analysis of computer simulations, are:

For a model in which competitive firms have to make their production decisions before observing the price of a single good in a market, the values of prices and quantities to which the algorithm converged correspond to rational expectations equilibria. The beliefs of all firms about how much to produce and offer for sale converge to the same value which is equal to the optimal quantities if the market price is known.

For an overlapping generations model with a constant stock of money (in which agents make decisions about how much to consume of a single, non-storable good in the first period of their life, without knowledge of the price of a good in a second period), the algorithm converges to the unique monetary equilibrium (the moving average learning scheme converges to the same equilibrium and the results from experimental overlapping generations environments are in the domain of attraction of this equilibrium). The agents form beliefs about the consumption in the first period as if they learned to maximize their utility functions and learned the correct price prediction.

For an overlapping generations model with constant deficit financed through seignorage (where agents make the same decision as in the previous model), the algorithm converges to the low-inflation-rate equilibrium (which is stable under least-squares learning scheme and which is the domain of attraction for the convergence of experimental economies). The genetic algorithm also converges to this equilibrium for the values of deficit and initial inflation rate for which the least-squares scheme diverges.

In a simple asset market model where a group of traders learn about the relationship between the price and the return on asset, using genetic algorithms to determine their estimates of the return, price and quantities traded converge to rational expectations price and quantities. Convergence of genetic algorithm to rational expectations equilibria was also obtained for the values of a stability parameter for which least squares do not converge.

Acknowledgments

I would like to thank Brian Arthur, John Holland, Robert Lucas, Ramon Marimon and John Miller for their helpful comments. I would especially like to thank Thomas Sargent and Michael Woodford for their invaluable support and guidance. Finally, I would like to thank the Santa Fe Institute for their hospitality during the seven-month period that I spent there, preparing my thesis proposal, as part of the their Economics Research Program. This program is funded by grants from Citicorp/Citibank and the Russell Sage Foundation and by grants to SFI from the MacArthur Foundation, the National Science Foundation (PHY-8714918), and the U.S. Department of Energy (ER-FG05-88ER25054).

INTRODUCTION

This paper describes four applications of genetic algorithms (GAs) to economic modeling of agents' learning. At this stage, these applications consist basically of an analysis of the systems' behavior through computer simulations, which is just a part of a larger research project that will also include the analysis of the limiting behavior of the systems (convergence properties, the rate of the accumulation of knowledge, stability of equilibria under this learning algorithm as a selection criterion in the case of multiple equilibria etc.) and the comparison of GAs performance to results from experimental economics.

The first application is to a model of competitive firms that learn to predict the price of their product and hence how much to supply.

The second and the third applications deal with an overlapping generations model of fiat money with constant money supply and with a constant deficit financed through seignorage respectively.

The fourth application essentially treats Bray's(1982) model of an asset market with both informed and uninformed agents.

These particular models were chosen because they offer an opportunity to compare GAs performance with that of other learning schemes considered by others (Bray (1982), Lucas (1986), Marcet and Sargent (1986),(1987), Woodford(1989)). They further provide the basis for comparing their patterns and behavior with the experimental results(Lim, Sunder and Prescott (1987), Marimon and Sunder (1988)).

GAs have been developed by John Holland, his colleagues and his students at the University of Michigan. They are search algorithms based on the mechanics of natural selection and natural genetics. They combine the survival of the fittest among string structures with a structured, yet at the same time randomized information exchange to form a search algorithm with some of the flavor of human search. In every generation, a new set of artificial creatures (strings) is created using bits and pieces of the fittest of the old and occasional creation of new parts. These algorithms have been theoretically and empirically proven to provide robust search in complex spaces.

My interest in GAs and the reasons why I find them more appealing than other algorithms for studying learning of economic agents fall into four main categories, the first being that GAs seem to be more realistic as models of human cognition, the second their advantages as a way of solving optimization problems, the third being that less preexisting competence required in respect to a specific problem is assumed, compared to other learning models in economics and the fourth being their ability to represent the decentralized character of learning in economics.

1. What distinguishes GAs from most of the other learning schemes is the parallel processing of information, competition among alternative rules, selection of those that perform better and the possibility of creation of new rules. Recent studies in cognitive psychology (see Rumelhart, McClelland and the PDP research group (1987) and Holland, Holyoak, Nisbett and Thagard (1986) show that parallel algorithms are probably better algorithms for the description of human cognition. Neurobiologists have been aware for quite a long time that brain neurons work in a parallel fashion. Recently, there has been the theoretical attempt to explain the functioning of the brain, not only at the level of neuronal

interaction, but also at the level of higher brain functions such as perceptual categorization, formation of short and long term memory and learning (see Edelman, 1987) by three basic mechanisms: parallelism, selection and reentry. Finally, the so-called thinking machines, which work in a parallel fashion (with 64000 processor units) are able of accomplishing tasks that resemble "true" learning much more than what serial computers (no matter how powerful) were ever capable of performing.

2. The advantages of GAs in solving optimization problems can be summarized in the following points:

Rather than working with parameters themselves, GAs work with a coding of the parameter set. It requires that the set of the optimization problem be coded as a finite-length string over some finite alphabet. By exploiting coding similarities in a very general manner, they become largely unconstrained by the limitations of other methods (continuity, derivative existence, unimodality and so on).

- As opposed to point-to-point methods of optimization, whose major inadequacy is the possibility of stopping at a merely local maximum in multimodal search space, GAs search from a population of points, and simultaneously climb many peaks in parallel, thus reducing the probability of stopping at a false optimum.

- GAs employ payoff (objective function) information, not derivative or other auxiliary information.

- In their search, GAs use probabilistic transition rules. Regions of the search space tested by GA in each iteration are chosen at random.

3. The models where agents use GAs as a learning scheme require less prior knowledge, compared to all other schemes used as learning algorithms of economic agents. All other models contain the assumption of the maximization of an agent's objective function, whereas such an assumption is not required within the GA framework. Instead, while learning about their environment, agents also learn how to maximize their profits or utility functions and finally, begin to maximize them in equilibrium.

4. Decentralization is a feature of the economic process in market economies and it may be important to how the convergence to equilibrium occurs in these economies. GAs offer the possibility of analyzing decentralized learning and studying whether it has properties different from those of centralized learning process, in which all agents have the same beliefs and update them in the same way in response to experience.

Two possible ways to think about a GA population of strings (chromosomes) are (1) that a single agent has a whole population of strings, where each string represents one of an agent's alternative ideas, or (2) that a population of strings represents a whole population of agents with different opinions, each string standing for an individual agent. I investigate models of both kinds, and some of my results could be given either interpretation.

In my applications of GA, fitness values depend upon the entire population in existence. As the environment is not fixed during learning, and since the behavior of the population determines the values of the variables, the fitness of each string depends on the overall behavior of the population. This distinguishes the economic applications that I am interested in from the kinds of optimization problems considered by Holland and co-workers.

Finally, I would like to mention that I have modified standard algorithms developed by Holland and co-workers to obtain better results for the kind of problems that I treat. The exact modifications are explained later in the text.

The paper is organized in five sections:

Section 1 contains a description of the structure and operation of GA, the basic tools for the analysis of their performance and an outline of previous GAs applications related to optimization and learning.

Sections 2 - 4 describe my applications of GAs to the above mentioned models. Each of these sections consist of three parts: the first depicts the basic model and describes equilibrium, the second contains a description of the other learning schemes that have been applied to the same type of model (it is included to make the comparison of these algorithms with GA behavior more comprehensible and self-contained), and in the third part I give the description of the particular GA computer simulations, of the results of these simulations together with the comparison to the results of the above mentioned learning algorithms.

Section 5 outlines my plans for a future research.

1. DESCRIPTION OF GENETIC ALGORITHM

Three - Operator Genetic Algorithm

The domain of the GA action is the set \mathcal{A} of populations of chromosomes. The set \mathcal{A} is a potential, rather than actual set, in the sense that the elements of the set become available to the algorithm only by successive modifications (not by selection from an extant set). The iterations of the algorithm last for T generations, where T is given by a designer of the problem.

A population of generation t , ($t = 0 \dots T$), denoted by $A(t)$ consists of n chromosomes denoted by $A_{i,t}$, $i = 1 \dots n$.

Each chromosome is a binary string consisting of 1's and 0's with l alleles (positions), where l is the length of a chromosome. An allele is denoted by $a_{i,t}^k$, where k is the position in a string $A_{i,t}$ that the allele occupies, $k = 1 \dots l$.

The modifications are performed using a set of genetic operators, Ω , that includes: reproduction, ω_r ; crossover, ω_c and mutation, ω_m .

Reproduction is the process in which individual strings are copied based on the value of their objective function (the fitness function), whereby the strings with higher fitness values are given higher probability of contributing offspring to the next generation. The algorithmic form of the reproduction operator is like a biased roulette wheel, where each string is allocated a slot sized in proportion to its fitness. It operates in such a way that the number of spins is equal to the number of strings in a population, where each spin yields a reproduction candidate. Once a string is selected, its exact copy is made. After the reproduction is completed, strings are entered into a "mating pool" for further genetic operator action.

Crossover is the randomized process of selection and exchange of parts of strings. Its primary role is generation of new, possibly better performing, structures. It operates in two stages: Firstly, two strings A and A' are selected from the mating pool at random.

Secondly, a number x is selected from $(1 \dots l - 1)$ again at random, where l represents the length of a string. Two new strings are formed by swapping the set of alleles to the right of the position x , yielding

$$a_1 \dots a_x a'_{x+1} \dots a'_l$$

and

$$a'_1 \dots a'_x a_{x+1} \dots a_l.$$

Mutation is the process whereby the value of a string allele is altered by a random process wherein each position has a small probability of undergoing mutation, independently of what happens at other positions.

Its primary role is to make sure that no allele permanently disappears from the population, since below average performance strings eliminated during reproduction might be required for improvement at some later stage.

GA produces a sequence of populations, i.e. a trajectory through \mathcal{A} , by selecting populations that perform better in a given environment. It receives a direct indication of the performance of each population it tries and of each string within a population through the payoff (objective or fitness) function: $\mu : \mathcal{A} \rightarrow R^n$. This function shows how well a particular string performs in a given environment. It can take different forms, which depends on the design of a specific problem.

In each generation t , the algorithm uses the following steps to modify members of a population, thereby producing a sequence of new populations for trial (the basic cycle given below is the one described in Goldberg, 1989):

1. Evaluation of a fitness value, $\mu_{i,t}$, for each member of the population of generation t .
2. Reproduction of strings' copies using the reproduction operator, ω_r . The probability that a string $A_{i,t}$ will get a copy, $C_{i,t}$, that undergoes further genetic operations is given by

$$P(C_{i,t}) = \frac{\mu_{i,t}}{\sum_{i=1}^n \mu_{i,t}}, \quad i = 1 \dots n.$$

3. Random selection of mating pairs, j and $j + 1$, $j = 1 \dots n - 1$.
4. With application of the crossover operator, ω_c to the selected pair of mates with the probability of crossover, p_{cross} and the mutation operator, ω_m , with the probability of mutation, p_{mut} , members of a new population of generation $t + 1$ are obtained.

$$A_{j,t+1} = \omega_m(\omega_c(C_{j,t}, C_{j+1,t}))$$

$$A_{j+1,t+1} = \omega_m(\omega_c(C_{j,t}, C_{j+1,t}))$$

The algorithm is iteratively repeated for T generations. The initial population, at $t = 0$, is generated randomly.

Holland's Schemata Theorem

Holland's Theorem of Schemata (1975) deals with the operation of GAs, their leverage in information processing and with their potential of creating better performing strings for solving problems in arbitrary environments.

The theorem exploits the wealth of information provided by the population of strings. It identifies the similarities among strings and introduces the notion of schema (building block) - high performing string. It further considers the effect of reproduction, crossover and mutation on a particular schema, quantifies its growth and decay rates and gives the mathematical form of these rates. Finally, it offers explanation how combining the building blocks leads to high performance of GA in arbitrary problems.

A schema is a string over an extended alphabet, $0, 1, *$, where the 0 and the 1 retain their normal meaning and the $*$ is the "don't care symbol".

It is a similarity template describing a subset of strings with similarities at certain predetermined string positions. There are 3^l schemata, defined over a binary string of length l and in a population of n strings there are at most $n2^l$ schemata contained in a population, because each string is itself a representation of 2^l schemata. Two important properties of schemata are their order and defining length. The order of schema ϵ , denoted by $o(\epsilon)$, is the number of fixed positions (the number of 1's and 0's) and the defining length of schema ϵ , denoted by δ , is the distance between the first and the last fixed string position.

If only reproduction is at work in GA, the expected number of schema of type ϵ in generation $t + 1$ is proportional to the ratio of that schema average fitness to the average fitness of the whole population in generation t , $E(N_{t+1}(\epsilon)) = \frac{\mu_t(\epsilon)}{\mu_t} N_t(\epsilon)$. Starting from the generation $t = 0$ and assuming a starting constant value of schema ϵ , it can be shown that reproduction allocates an exponentially increasing (decreasing) number of trials to above (below) average schemata.

If crossover occurs, with probability p_{cross} for a particular mating, the survival probability p_{sc} of schema ϵ is given by: $p_{sc} \geq 1 - p_{cross} \frac{\delta(\epsilon)}{l-1}$. Short, low-order schemata have lower probability of being disrupted by crossover.

Particular schema survives mutation if each of its $o(\epsilon)$ positions are left unchanged. Its probability of surviving mutation is given by: $p_{sm} \geq (1 - p_m)^{o(\epsilon)}$. Since the mutation takes place with very small probability, $p_m \ll 1$, p_{sm} can be expressed by: $1 - p_m o(\epsilon)$.

Therefore, the combined effect of reproduction, crossover and mutation on the expected number of copies of ϵ in generation $t + 1$ is expressed by:

$$E(N_{t+1}(\epsilon)) \geq N_t(\epsilon) \frac{\mu_t(\epsilon)}{\mu_t} \left[1 - p_{cross} \frac{\delta(\epsilon)}{l-1} - o(\epsilon)p_m \right]$$

The above formula expresses the basic statement of the Schema Theorem: Schemata of short length and low order, with above average performance are sampled at exponentially increasing rates. In other words, instead of trying every possible combination in a population of n strings, better and better strings are constructed from the best partial solutions of past samplings. The population of n strings serves as a database that compactly and usefully summarizes the information obtained from the search. The process of propagating

a large number of building blocks from generation to generation done in parallel is called intrinsic (implicit) parallelism. Holland's estimate (1975) of effectively processed schemata in a population of n strings is of the order n^3 .

Applications of GAs

Due to their relatively simple structure and operation and their superiority in comparison to standard search techniques, GAs have attracted wide attention in various disciplines, such as biology, computer science, engineering, social sciences, medicine, mathematics etc.

Early applications have introduced a lot of new mechanisms and genetic operators aimed at improving the performance and speed of GAs. While these early applications tried to perform simulations of genetic systems and resembled function optimization, it remained largely unclear that natural search algorithms could successfully be used in artificial settings. Consequently, experiments designed to reiterate the significance of individual mechanisms have been rarely conducted. The most notable early applications include:

- game of hexapawn - Bagley(1967) - GA was used to search for parameter sets in game evaluation function in comparison to correlation algorithms;
- pattern recognition - Caviccio (1970) - GA was used in the architecture of a pattern recognizing machine for the image storage and association with class names, and for matching the unknown image with the stored class during recognition stage.

More carefully controlled experiments started only after the development of Holland's theory of schemata, which unshrouded much of the complexity of earlier GAs and underlined the relative importance of reproduction and crossover, as well as the secondary role of mutation in artificial genetic search. His theory also emphasized the significance of structural recombination for the implicit parallelism.

Further advancement in terms of improvement of operation of GAs came with function optimization works of Hollstein(1977) and particularly with De Jong's dissertation (1975). De Jong devised two measures to quantify the effectiveness of GAs, which served to measure convergence and on-going performance.

Current application of GAs include:

- engineering - Goldberg (1983) - GA was applied in natural gas pipeline control project;
- medical image registration - Grefenstette, Fitzpatrick and Van Gucht(1984) - simple GA was used to perform image registration as a part of larger digital angiography system;
- structural optimization - Goldberg and Samtani (1986) - GA was applied to optimize the weight of different structures under pre-set constraints in civil engineering projects.

One of the very promising applications of GAs is their role in classifier systems. These are the machine learning systems which learn syntactically simple rules encoded as bit strings, and evolve on the basis of intermittently given stimuli and reinforcement from its environment. The interactions of rules and GA lead to a system that evolves to maximize positive reinforcement from the environment. The role of GA within the framework of classifier systems is to enable them to choose new, possibly better rules and thus improve their performance in a changing environment.

There have also been a few earlier applications of GAs and classifier systems in economics:

game theory

- Axelrod (1985,1987) designed a GA strategy that was able to outperform "tit for tat" in a repeated Prisoner's Dilemma game, although "tit for tat" previously won among 76 different computer strategies that were competing in the same game.

- Miller (1988) simulated the repeated Prisoner's Dilemma game, under the conditions of perfect and imperfect information, in which players' strategies, submitted in the form of finite automata, were modified using GAs; results of these simulations show that higher levels of imperfect information are associated with less cooperation and lower payoffs.

economic optimization problems

- Miller (1986) simulated models of consumer demand behavior, behavior under uncertainty and of market structure in which agents learned to optimize their objective functions.

classifier systems

- Marimon, McGrattan and Sargent (1989) studied exchange economies of Kiyotaki and Wright model in which agents must use a commodity or fiat money as a medium of exchange if trade is to occur; their "artificially" intelligent agents learn their trading and consumption strategies adaptively, using classifier systems; GA is introduced to study the evolution of the economy when rules are randomly generated; for most of the simulated economies, trading and consumption patterns converge to a stationary Nash equilibrium, even if agents start with random rules.

- Brian Arthur, John Holland and Richard Palmer of the Santa Fe Institute are constructing a stock market model using classifier systems.

2. MODEL OF COMPETITIVE FIRMS THAT LEARN TO PREDICT THE CORRECT PRICE

There are n firms in a competitive market that are price takers and that produce the same good.² Since the production takes time, quantities produced must be decided before market price is observed.

First, I am going to describe a rational expectations version of the model.

The cost of a production of a firm i is:

$$C_t^i = xq_t^i + \frac{1}{2}y(q_t^i)^2 \quad (1)$$

where C_t^i is a firm i 's cost of production for sale at time t and q_t^i is a quantity it produces for sale at time t .

The profit of an individual firm is:

$$\Pi_t^i = P_t q_t^i - xq_t^i - \frac{1}{2}y(q_t^i)^2 \quad (2)$$

At $t - 1$, each firm chooses a quantity, q_t to maximize $E_{t-1}\Pi_t^i$ on the basis of their expectations about price, P_t , given by

$$E_{t-1}P_t = x + yq_t^i \quad (3)$$

The price P_t that prevails in the market at time t is determined by the demand curve:

$$P_t = A - B \sum_{i=1}^n q_t^i \quad (4)$$

In rational expectations equilibrium $E_{t-1}P_t = P_t$, i.e. firms' expectations about the price, P_t , of a good in period t equal the equilibrium price (Muth (1961)). Thus,

$$x + yq_t = A - Bnq_t$$

$q_t = q_t^i$ for all i , or

$$q_t = q^* = \frac{A - x}{bn + y} \quad (5)$$

The objective of the GA application is to see whether quantities produced and offered for sale by firms that are using GA as their learning scheme converge to this constant quantity, q^* .

The genetic algorithm works as follows:

1. The population of chromosomes in generation t , which are strings of finite length of 0's and 1's, represents the production decision rules of individual firms. The number of

² Marcet and Sargent, (1989) study a stochastic version of a very similar model. The economic structure of their model is a version of Frydman's(1982) model. In their model firms use least squares to predict a price.

strings is equal to the number of firms in the market. A decoded and normalized value of each binary string, a member of a population in generation t , represents a firm's belief about a quantity that a firm decides to produce in time period t (generation t).

2. For a string of length l the decoding works in the following way:

$$x_k = \begin{cases} x_{k-1} + 2^{k-1}, & \text{for } a_{i,k} = 1 \\ x_{k-1}, & \text{for } a_{i,k} = 0 \end{cases}$$

for $k = 1 \dots l$ and where $a_{i,k}$ is the k^{th} allele in a chromosome and x_k is a real number representing the decoded string value up to allele k . After a string is decoded, x_l is normalized. In general, a normalization coefficient depends on a particular GA application:

$$q_t^i = \frac{x_l}{normc}$$

where $normc$ is a coefficient chosen to normalize the value of x_l . Decoded and normalized values of strings correspond to firms' beliefs in regard to the quantity they should produce and offer for sale on the market.

3. After the determination of quantities that each firm produces and offers for sale, all offered quantities are summed and the market price of generation t , P_t , is computed, using (4).

4. Costs associated with produced quantities are computed for each firm, using (1).

5. Using the price, P_t of generation t , quantities that firms offered on the market and their production costs, a profit for each individual firm is then calculated, using (2).

6. Profits represent fitness values of individual chromosomes.

$$\mu_t^i = \Pi_t^i \quad i = 1 \text{ to } n$$

Fitness determines the probability of reproduction. The higher the profits, the higher the chance that a chromosome will get one or more copies in a next generation, $t + 1$.

7. Genetic operators reproduction, crossover and mutation take place and a population of a new generation is determined.

8. Steps 1-7 are applied iteratively, starting from a randomly generated initial population (generation 0).

The corresponding economic interpretation of the whole process may be the following: The reproduction works like imitation of successful rivals, where the production decision rules of the firms whose beliefs are given by well performing strings are copied by others, since these are the firms that earn higher profits in the market. Strings with lower fitness values, which means worse production decisions and lower profits, get less copies (or none) in the next generation as investors or financial intermediaries are not willing to allocate investment funds into an unprofitable production. Crossover and mutation are used to generate new ideas (beliefs) on how much to produce and offer for sale, recombining existing beliefs and generating new ones with crossover and mutation.

Using the algorithm given in Goldberg (1989), I have encountered difficulties with its convergence. The algorithm would not settle to the particular value, no matter how many generations a simulation was run for. It would get close to the unique rational expectations

equilibrium, but then continue to fluctuate around it without settlement. The explanation for this behavior is the following: although things improve rapidly at the beginning of a simulation (chromosomes have rather high rate of learning, their fitness values (profits) increase fast), after this initial phase of fast learning and improvement, each time new copies of chromosomes are reproduced and crossover and mutation take place, there is a chance that a "bad" crossover or mutation will affect a chromosome and instead of improving, decrease its fitness. When the algorithm gets to the optimum, there is nothing to prevent the possibility of leaving the optimum by a finite length.

The crossover operator can affect the convergence of the algorithm in generations in which different strings still exist in a population. For example, suppose that in a population at generation t one string whose decoded value corresponds to the solution to a search problem is created. In that case, there is a chance that a genetic material of that string which has reached optimum gets exchanged with the genetic material of a string still relatively far away from the optimum in such a way that both new offsprings, resulting from this combination, have fitness values lower than the best fit parent's. Since offsprings replace their parents in a new population, good genetic material can be lost, at least temporarily.

If all strings in a population are identical, the crossover cannot change anything. Repeated application of the crossover operator on such a population leaves the population unchanged.

On the other hand, mutation continues affecting the population as long as the algorithm is running. Since it changes a string bit value with a given probability, it can change an allele value of a string that has already reached the optimum and place it further away from the maximum (or minimum) by the amount equal to $2^{k-1}/normc$ where k is the position of an allele that undergoes mutation ($k = 1 \dots l$, where l is the total length of a string).

In order to overcome this problem, I have included the procedure that tests the fitness of an offspring before it is allowed to enter into a new population. (In the initial algorithm, offsprings automatically replace their parents in a new population). The exact extension of the algorithm is: After the creation of a couple of offsprings is over (i.e., after reproduction, crossover and mutation have taken place) the fitness values of the offsprings are computed. In the case of the competitive firms model, this means that the offsprings' string values are decoded as quantities produced and offered and the price of the previous generation is used to compute their profits, i.e., fitness values of newly generated offsprings. Then, fitness of each offspring is tested against the fitness values of each of the parents. If its fitness is higher than the fitness of one of the parents or of both of them, the offspring enters into a population of a new generation:

- if only one (out of two offsprings for each parents' pair) has a fitness higher than both of its parents, it replaces a parent with a lower fitness and a parent with a higher fitness remains in the population;

- in the case that both offsprings have fitnesses higher than a fitness value of each parent, they replace both parents as new members of the population;

- if both parents have fitnesses higher than their offsprings, they remain in the population of the new generation.

The above mentioned technique that I used enables the sort of an endogenous shut off of crossover and mutation once an algorithm is close to convergence.

The economic interpretation of this addition to the original algorithm might be the following: in each time period (each generation) firms generate new production decisions using genetic operators. They compare the fitnesses of these new potential proposals to the old set, under the market conditions observed in the past. Only new ideas that appear promising on such grounds are actually implemented (although the generation of new ideas is random their implementation is not).

Note that the above described procedure does not reduce GA search possibilities. Every point in a search space that would be sampled without this procedure is actually tested before determining whether it enters a new population as an encoded string.

In each simulation, the size of a population was either 20 or 30. Some application in artificial intelligence that studied the behavior of the algorithm in the optimization of various types of functions (De Jong (1975,1980)) showed that the best performance of the algorithm was achieved using the population of 30 chromosomes with the length of a chromosome equal to 30 bits.

In each of the simulations, the values of prices and quantities to which the algorithm converged correspond to rational expectations equilibrium. All the strings in a GA population of each simulation became identical, i.e. the beliefs of all firms about how much to produce and offer for sale converge to the same value which is equal to the optimal quantities if the market price is known. Results of the simulations are given in table 1. Graph 1 shows the time path of prices and quantities for the set of parameter values: $A = 100$, $B = 0.02$, $x = 30$ and $y = 1$.

It is worth noting that with the genetic algorithm learning described above, individual firms do not use first-order conditions for decision making. They do not equate marginal cost to the expected price and need not calculate either in order to decide how much they are going to produce in the following period. Still, by the time the algorithm converges, firms have learnt not only how to predict the correct rational expectations equilibrium price, but also how to make production decisions that will maximize their profits.

3. OVERLAPPING GENERATIONS MODEL

Economic models with incomplete market structure in which money serves the role of overcoming the constraints imposed by limited exchange possibilities usually have a continuum of equilibria. The overlapping generations model (OLG) Samuelson (1958), Wallace(1980), of a monetary economy is one such example of an environment which involves multiplicity of perfect foresight or rational expectations equilibrium paths. Whereas the rational expectations hypothesis does not characterize behavior outside equilibrium paths, hypothesis of the adaptive behavior on the part of agents who have to learn about their economic environment does provide guidance on how agents may behave under any observed history.

As Lucas(1986) has suggested, the stability results obtained through the analysis of learning dynamics in these models can be used to single out more likely equilibria, since these dynamics may be a plausible conjecture about actual human behavior. Given that GAs are very good global search algorithms and given their processing power there is a possibility that they approximate patterns of human behavior in experimental overlapping generations economies better than other learning schemes.

While, as Lucas (1986) points out, the adaptive behavior of agents characterized by using standard, enumerative (point by point) learning schemes is not based on any economic principle, beliefs that agents form using GA learning scheme are subject to competition with the survival of only those that result in better performance of agents in a given economy.

Furthermore, using some of the advanced genetic operators (inversion, dominance), it is possible to study the behavior of a system when policy changes are introduced. The resulting patterns of GA behavior should be much more interesting than those that could be obtained using standard learning algorithms. This is the part of my future research which is explained in more detail in the last section.

At the initial stage of my study of learning in OLG models, I have started with the simulations of learning in two different OLG environments:

- a) constant money supply and
- b) constant deficit financed through the seignorage.

3.1. Model with constant money supply

This is a standard OLG model with fiat money. Each generation consists of an equal number, n , of agents. Every agent belonging to generation t lives over two consecutive periods, t and $t+1$, and consumes c_t^1 in the first period (young age) and c_t^2 in the second period (old age). Each of them is endowed with e^1 units of a consumption good at the beginning of the young age and e^2 units of a consumption good at the beginning of the old age; $e^1 > e^2$.

The economy has nM^s units of money, M^s units per person, initially held by the agents whose last period of life is in period zero (first period of the model).

Each individual's preferences are given by $U(c_t^1, c_t^2)$ and the utility maximization problem is:

$$\begin{aligned} & \max U(c_t^1, c_t^2) \\ & \text{s.t. } m_t = (e^1 - c_t^1)P_t \\ & \quad c_t^2 = e^2 + \frac{m_t}{P_{t+1}} \end{aligned}$$

where m_t is the amount of money an individual acquires in the first period and spends in the second.

In my simulations I use the utility function

$$U(c_t^1, c_t^2) = c_t^1 c_t^2$$

The first order conditions for maximization of this problem are:

$$\frac{(e^2 + \frac{m_t}{P_{t+1}})}{P_t} = \frac{(e^1 - \frac{m_t}{P_t})}{P_{t+1}}$$

In equilibrium, the money demand per capita, m_t , must equal the supply, M^s , in each period. Hence, we can substitute M^s for m_t in the above:

$$P_{t+1} = \frac{e^1}{e^2} P_t - \frac{2M^s}{e^2} \quad (1a)$$

The stationary solution to this difference equation represents equilibrium in which a version of quantity theory of money holds. It is given by $P_t = P_s$ for all t , where

$$P_s = \frac{2M^s}{e^1 - e^2} \quad (2a)$$

This stationary competitive equilibrium price system exists for $\frac{e^1}{e^2} > 1$. It is the only stationary equilibrium with valued fiat money.

There is also a continuum of monetary equilibria indexed by initial price levels P_0 , in the interval (P_s, ∞) . All of the equilibria with an initial price greater than P_s converge to situations in which money has no value. This has led some to suppose that the stationary equilibrium with valued money is "unstable", and so, unlikely to be reached. The results of the learning analysis presented below provide an interesting contrast with this view.

Without valued currency there fails to exist a Pareto optimal equilibrium. Monetary, stationary equilibrium, in which $c^1 = c^2$ is Pareto optimal, and is also the stationary allocation that maximizes the stationary level of utility.

Moving Average Learning Scheme

Lucas (1986) studies the infinite horizon OLG economy with constant money supply in which agents use a moving average adaptive scheme to form expectations about the

next period price level. He shows that the algorithm converges to the stationary monetary equilibrium analogous to the one described above.

Lucas considers a model like that described above, except that preferences in this model are given by:

$$U(c_t^1, c_{t+1}^2) = (c_t^1)^{1/2} + 2(c_{t+1}^2) \quad (I)$$

In this case (1a) becomes

$$P_{t+1} = 4P_t \left(1 - \frac{M_s}{P_t}\right)^{1/2} \quad (II)$$

and the stationary monetary equilibrium is $P_s = \frac{16}{15}M_s$.

If, instead of rational expectations, agents form their expectations about the price in the next period adaptively, using the moving average learning scheme to form their price expectations, (II) becomes

$$P_{t+1}^e = 4P_t \left(1 - \frac{M_s}{P_t}\right)^{1/2} \quad (III)$$

where P_{t+1}^e is a point expectation formed at t about the price in $t+1$. The rule that agents use to update their expectations is given by

$$\frac{1}{P_{t+1}^e} = \frac{t}{t+1} \frac{1}{P_t^e} + \frac{1}{t+1} \frac{1}{P_{t-1}} \quad (IV)$$

which is equivalent to

$$\frac{1}{P_{t+1}^e} = \frac{1}{t+1} \left[\frac{1}{P_t} + \dots + \frac{1}{P_0} \right] \quad (V)$$

The dynamics defined by (III) and (IV) result in sequences $\{\frac{1}{P_t}\}$ and $\{\frac{1}{P_t^e}\}$ that satisfy

$$\lim_{t \rightarrow \infty} \frac{1}{P_t} = \lim_{t \rightarrow \infty} \frac{1}{P_t^e} = \frac{15}{16} \quad \text{for all } \frac{1}{P_1^e} = (0, 1)$$

The system with the moving average adaptive scheme converges to the monetary stationary equilibrium.

Genetic Algorithm Application

As in the previous model, each string in a population of chromosomes represents a decision rule of a single individual. Each chromosome is a belief of an agent about how much he should consume in the first period of his life. Decoding and normalization are done in the same way as in the first model. A decoded and normalized string value gives his first period consumption.

The reason for choosing this particular representation is related to the work done with the experimental OLG economies. Lim, Prescott and Sunder(1987) conducted experiments in the OLG environment with constant money supply and Marimon and Sunder(1988) in

the environment of OLG model with deficit financed through seignorage. In both of these studies, the participants of the experiments were able to make decision about their first period consumption. Having the same GA setup gives me the opportunity to compare the experimental results with the GA performance.

The following notation and formulas are used:

M^s - money supply per head - initially distributed to the old members of generation $t = -1$

n - number of strings

maxgen - total number of generations

c_{it}^1 - decoded and normalized value of a string

$S_{it} = e^1 - c_{it}^1$ - individual savings

$m_{it} = S_{it}P_t$ - individual nominal money holdings at the end of the first(young) period

$c_{it}^2 = \frac{S_{it} * P_t}{(P_{t+1})} + e^2$ - individual consumption in the second (old) period

$U_i(c_{it}^1, c_{it}^2) = c_{it}^1 c_{it}^2$ - fitness of an individual string

The price of the consumption good is given by:

$$P_t = \frac{nM^s}{\sum_{i=1}^n S_{i,t}} \quad (3a)$$

for $t = 0 \dots \text{maxgen}$.

The algorithm works as follows:

- In each generation t , there are two populations of chromosomes, one being the new population of generation t and the other the population of generation $t - 1$. Since the fitness values of strings can be determined only after the total life cycle of two periods, I had to work with two populations of chromosomes that were reproduced every other period.

- Bit strings of generation t are decoded and normalized. These normalized values are the values of consumption of members of generation t . Their savings, $S'_{i,t}$ s, are computed and the price P_t is calculated and then, the consumption in period two, for members of generation $t - 1$ is determined:

$$c_{i,t-1}^2 = \frac{S_{i,t-1}P_{t-1}}{P_t} + e^2; \quad (4a)$$

- Fitness values of the members of generation $t - 1$ are computed.

-Population for generation $t + 1$ is generated from the population of generation $t - 1$, using genetic operators reproduction, crossover and mutation; final members of the population are determined by the comparison of the fitness values of the members of the old population and the fitness values of the newly generated strings. The offsprings' fitness values that are used for comparison with their parents' fitnesses are calculated using their decoded string values and the actual prices from the two previous periods. Whether or not a generated offspring enters into a population of a new generation is determined in the way described in the algorithm of the competitive firms model.

Once the population of new generation $t + 1$ is created, the whole cycle is repeated. Consumption and savings values for the members of generation $t + 1$ are computed, while the genetic operators are performed on the members of generation t .

The populations of generations 0 and 1 are randomly generated. The system starts off with nM^s units of money distributed to initially old that use all of it for the demand of the consumption good.

I ran the simulations for four different sets of endowment values. The first set was : $e^1 = 150, e^2 = 10$, the second set was: $e^1 = 120, e^2 = 20$, the third set was: $e^1 = 100, e^2 = 90$ and the fourth set was $e^1 = 7, e^2 = 1$. The fourth set corresponds to the endowment pattern of OLG experimental economies with constant money supply. Lim, Sunder and Prescott (1987).

The size of the population of chromosomes was 30. The length of the chromosome string was 30 bits.

Each simulation was run for 200 generations. The results are given in table 2 (models I, II, III and IV). They show that, in each of the simulations, the algorithm converged to the stationary competitive equilibrium, which is Pareto optimal, the same equilibrium to which the moving average scheme converges.

The beliefs about the amount that should be consumed in the first period in both populations converge to the same value, so that, after the algorithm converges, the members of both populations form beliefs about the consumption in the first period as if they learned to maximize their utility functions and learned the correct price prediction.

3.2. Model with constant deficit financed through seignorage

The description of the economy of this model is the same as the one of the previous model in the part concerning agents' length of life, number of agents, endowment pattern and preferences. The difference is that, now, instead of keeping money supply constant, government finances deficit per head, d , through seignorage.

Equation (1b) again describes equilibrium except that now the money supply in period t (M_t^s units per head) is no longer constant. As a result we have:

$$M_t^s = P_t S_t = 1/2(e^1 - \pi_{t+1} e^2) P_t \quad (1b)$$

where $\pi_t = P_{t+1}/P_t$.

The monetary policy is given by:

$$d = \frac{M_t^s - M_{t-1}^s}{P_t} \quad (2b)$$

or

$$d = \frac{e^1}{2} - \pi_{t+1} \frac{e^2}{2} - \frac{e^1}{2\pi_t} + \frac{e^2}{2}$$

Under the rational expectations hypothesis, paths of equilibrium inflation rates are given by

$$\pi_{t+1} = \frac{e^1}{e^2} + 1 - \frac{2d}{e^2} - \frac{e^1}{e^2} \frac{1}{\pi_t} \quad (3b)$$

This equation has two stationary solutions, higher and lower inflation rate. The higher one is the stable solution characterizing the asymptotic behavior of a continuum of rational expectations equilibrium paths.

Least Squares Learning Scheme

Marcet and Sargent (1987) analyze the least squares learning scheme in the context of the Sargent's and Wallace's (1985) model of hyperinflation. The equilibrium in this model, stable under the rational expectations hypothesis, is the high inflation rate, stationary equilibrium. Marcet and Sargent show that, under least squares learning, the model either converges to the low inflation, stationary equilibrium or no equilibrium exists. Furthermore, for some high levels of the deficit for which there exist an equilibrium under rational expectations, there is no equilibrium under least squares learning. This result is also classical in the sense that a higher permanent deficit is associated with a higher stationary inflation rate, while under rational expectations, a higher permanent deficit results in a lower stationary inflation.

The corresponding least squares learning algorithm can be applied to the above described OLG model.

We can see from equations (1b) and (2b) the evolution of prices and money holdings which is the following:

$$P_t = \frac{e^2}{e^1} P_{t+1} + \frac{2}{e^1} M_t^s \quad (I)$$

$$M_t^s = M_{t-1}^s + dP_t \quad (II)$$

Instead of having rational expectations about the level of prices, agents learn about it using a least squares adaptive scheme. Their expectation of the price in $t + 1$ is

$$E_t P_{t+1} = \beta_t P_t$$

where β_t is the estimate of the inflation rate obtained through the regression on past values of prices:

$$\beta_t = \left[\sum_{s=1}^{t-1} P_{s_1}^2 \right] \left[\sum_{s=1}^{t-1} P_s P_{s-1} \right]$$

With least squares learning the evolution of inflation rates is given by:

$$\beta_t = (1 - g_{t-1})\beta_{t-1} + g_{t-1} \frac{(1 - \frac{e^2}{e^1}\beta_{t-2})}{(1 - \frac{e^2}{e^1}\beta_{t-1})} S(\beta_{t-1}) \quad (III)$$

or approximately by,

$$\beta_t = (1 - g_{t-1})\beta_{t-1} + g_{t-1} S(\beta_{t-1}) \quad (IV)$$

where

$$g_{t-1} = \frac{P_{t-2}^2}{R_0 + \sum_{j=1}^{t-1} P_{t-1}^2}$$

$$S(\beta) = \frac{1 - \frac{e^1}{e^2}}{1 - \frac{e^1}{e^2} - \frac{2}{e^1}d}$$

and R_0 is an initial value. The fixed points of the least squares map, β_1^* and β_2^* , are equal to the fixed points of the map $S(\beta)$ which is the map of the time invariant learning rule. The two fixed points of $S(\beta)$ correspond to π_1^* and π_2^* .

Marcet and Sargent show (proposition 2) that in the difference equation (IV), β_t converges to β_1^* for any $\beta_0 = (0, \beta_2^*)$, given that $0 < g_{t-1} < 1$. If $\beta_0 > \beta_2^*$, β_t will get to the region where no equilibrium exists.

In equation (III), $\{\beta_t\}$ converges to β_1^* , provided that

$$\left(1 - \frac{e^2}{e^1}\beta_{t-2}\right) / \left(1 - \frac{e^2}{e^1}\beta_{t-1}\right) S(\beta_{t-1}) < \beta_2^*.$$

However, if this term is greater than β_2^* and if g_t is big enough, the situation in which $\beta_t > \beta_2^*$ and in which the path diverges away from β_1^* , toward the region where no equilibrium exists, can occur. When $\{\beta_t\}$ is converging to β_1^* from below, $\frac{1 - \frac{e^2}{e^1}\beta_{t-2}}{1 - \frac{e^2}{e^1}\beta_{t-1}}$ will be greater than unity and if g_t is sufficiently large, $\{\beta_t\}$ will diverge away from the least squares stable, low inflation rate equilibrium.

Application of Genetic Algorithm

The algorithm is the same as for the overlapping generations model with constant money supply, except for the way in which a price is computed in each time period t , because of the different monetary policy. Since government finances constant deficit per head, d , the price in period t is given by:

$$P_t = \frac{\sum_i^n S_{i,t-1} P_{t-1}}{\sum_i^n S_{it} - nd} \quad (4b)$$

for $\sum_i^n S_{i,t-1} > nd$. I tried four different endowment patterns in the first set of simulations. In all four simulations the algorithm converged to (or close to) the lower inflation rate stationary equilibrium. The results are given in table 3 (models I, II, III and IV). Graphs 2a and 2b show the evolution of inflation rates and consumption ratio for the set: $e^1 = 100$, $e^2 = 90$, $d = 0.007695$ and initial $M^s = 1000$.

I used the results from Marcet and Sargent (1987) as the guidance to what parameter values may be interesting for the second set of simulations and which would make it possible to compare the behavior of the genetic algorithm to the behavior of the least squares learning scheme.

The possible failure of convergence of β_t to β_1^* is more likely to happen when β_2^* is sufficiently close to β_1^* . The stationary gross inflation rates, β_1^* and β_2^* , approach the common value $b^{1/2}$ as the deficit, d , approaches from below the maximal feasible value of

$$d_{max} = \frac{e^2}{e^1} \left[1 + \frac{e^1}{e^2} - 2 \left(\frac{e^1}{e^2} \right)^{1/2} \right]$$

In all of the simulations $b = \frac{e^1}{e^2}$ (which corresponds to λ^{-1} in Marcet and Sargent) was set to 1.1111.

Depending on the values of deficit and initial inflation rates, the results of the least squares simulations were the following:

- a) For the values of deficit equal to 0.0019 the least squares converged to β_1^* .
- b) For the values of deficit equal to 0.0234 converged to β_1^* if the initial inflation was close to β_1^* and diverged otherwise.
- c) For the values of deficit equal to 0.0024, the least squares algorithm diverged, oscillating.

The low inflation stationary equilibria for these cases are:

- a) $\beta_1^* = 1.026$
- b) $\beta_1^* = 1.0357$
- c) $\beta_1^* = 1.0377$.

I ran two sets of simulations, with three simulations in each which, with their parameter values, correspond to three cases of Marcet and Sargent. The endowment pattern in all simulations was: $e^1 = 100$, $e^2 = 90$, which corresponds to $\lambda = 0.9$ in Marcet and Sargent (λ is equal to e^2/e^1). Deficit per head was set to values that correspond to the values of deficit, ϵ , in Marcet and Sargent. This time initial populations were not created randomly for I needed to do the simulations for specific initial rates of inflation that equal those used by Marcet and Sargent. All of these initial rates had to be below β_1^* since these are the values of initial inflation rates for which, depending on an initial rate itself and on the value of a deficit, (III) might not converge.

The difference between the parameter values of the two sets of simulations is the extent of the diversification of initial population. The members of the populations in the first set of populations were much more similar to each other than was the case in the second set.

The results of the first set of simulations are:

- for the case a), GA converged to the low inflation stationary equilibrium with the consumption ratio equal to β_1^* . after 670 generations;³
- for the case b), GA converged, after 350 generations, to the values 1.0389 for the inflation and 1.04158 for the consumption ratio.
- for the case c), monetary economy broke down.

The results of the second set of simulations are:

- for all three cases, a), b) and c) GA converged to, or very close, to the low inflation stationary equilibrium.

In all three simulations of the second set the algorithm starts out with relatively large fluctuations round β_1^* which eventually (round generation 200) become smaller and smaller. It remains fluctuating almost until it converges and it converges to β_1^* from above. In simulations for the cases a) and c) the convergence was achieved after generation 200 and for the case b) after generation 230.

The explanation for the results of the first set of simulations is that the algorithm did not start with sufficiently diversified populations. Since strings were very similar to each

³ Each simulation was run for 1000 generations. Convergence after a certain generation means that the algorithm remained at these values until the end of a simulation.

other, crossover could not do much to generate an improvement of the algorithm. Most of that task was left to mutation whose effect on the algorithm is relatively small, given its very low probability of occurrence.

Since most of the task that should be performed by crossover was left to mutation, that could be the reason for the extremely slow convergence in the case a) and the reason why the algorithm did not converge to the rational expectations equilibrium in the case b). In the case c), once everyone was consuming what he was endowed with in each period (all the strings became the same) crossover could do nothing to generate new faith in money. Mutation did continue to create some change, but since potential new populations had been tested against data of the previous two time periods (which is data of the economy in which no trade takes place) before they were allowed to enter into new generations, newly created strings, that are different and express willingness for trade, are going to get zero fitness values and will not be allowed to enter into new generations.

I intend to experiment more with different extents of initial beliefs diversification and different rates of mutation and examine how they affect convergence.

These results indicate that the stability conditions for the convergence of GA and those for the convergence of least squares in this model are probably different.

I have also run simulations with endowment patterns equal to those of Marimon and Sunder's experimental OLG economies with seignorage-financed deficit and I have started work on comparing the two patterns of behavior.

4. MODEL OF ASSET TRADING BY INFORMED AND UNINFORMED AGENTS

This section describes the application of GA to Margaret Bray's (1982) model of asset trading by informed and uninformed agents.

There are N_u uninformed traders and N_i informed traders in the market where a risky asset is traded. The return on the asset, r and the exogenous supply of the asset, X^s , are independently and normally distributed random variables.

Both types of agents, informed and uninformed, observe the price of the asset P_t at t and r_t at some time between t and $t + 1$. Informed agents also observe X_t^s at t and they know all the moments of the joint distribution of these variables in equilibrium, so that their expectations are rational. Uninformed agents are disadvantaged not only in that their information set is smaller, but that they are assumed to have to learn about the relation between P_t and r_t from observations.

The demand of an informed agent for the risky asset is given by:

$$D_i = N_i \theta_i [E_i(r_t | X_t^s, P_t) - P_t] \quad (1)$$

where

N_i is number of informed agents;

θ_i is coefficient of informed agents demand and E_i denotes the true conditional expectation.

The demand of uninformed agents for the risky asset is given by:

$$D_u = N_u \theta_u [E^u(r_t | P_t) - P_t] \quad (2)$$

where

N_u is number of uninformed agents;

θ_u is coefficient of uninformed agents demand;

$E_{t-1}^u(r_t | P_t) = a_{t-1} + b_{t-1} P_t$ denotes the forecasting rule used by uninformed agents at time t .

In Bray's model a_{t-1} and b_{t-1} are obtained from the least squares regression of r_s on P_s for $s = 1 \dots t - 1$.

The price of the asset in period t is determined by market clearing, which implies:

$$P_t = N_i \theta_i E^i(r_t | r, X_t^s, P_t) + N_u \theta_u E^u(r_t | P_t) - X_t^s / N_i \theta_i + N_u \theta_u \quad (3)$$

In order to analyze the stability of the least squares learning algorithm (i.e. convergence to the rational expectations equilibrium values of the intercept and the slope), Marcet and Sargent (1987) derive a mapping $S(a, b)$, given by

$$S(a, b) = [E(r_t) - kE(x_t)] / (N_u \theta_u) - ka, k(N_i \theta_i + N_u \theta_u) / N_u \theta_u - kb] \quad (4)$$

where

x_t is a random variable equal to $N_i \theta_i E(r_t | X_t^s) - X_t^s$,

$$k = \frac{N_u \theta_u}{[N_i \theta_i - \frac{1}{\rho}]}$$

and

$$\rho = \frac{\text{cov}(r_t, X_t^s)}{\text{var} X_t^s}.$$

A fixed point (a^*, b^*) of $S(a, b)$ is a rational expectations equilibrium. Differentiating the right side of (4), evaluating at (a^*, b^*) and applying proposition 1 (see Marcet and Sargent (1987)), Marcet and Sargent establish that the model is stable if and only if $k > -1$. Whether the inequality is satisfied or not depends on the values of the parameters of the model:

- if $1/\rho < 0$, then the stability region includes all $(N_i\theta_i, N_u\theta_u) \in R_+^2$
- if $1/\rho > 0$, then the stability region includes all $(N_i\theta_i, N_u\theta_u)$ satisfying $0 < N_i\theta_i < 1/\rho$ and $N_u\theta_u + N_i\theta_i > 1/\rho$

For any given pair (θ_i, θ_u) and $1/\rho > 0$ and any positive ratio N_i/N_u of informed to uninformed agents, there exist an absolute number of traders $N_u + N_i$ sufficiently small to guarantee stability of the learning mechanism and there also exist a number of traders sufficiently large to guarantee local stability; but for intermediate number of traders the rational expectations equilibrium is unstable.

Genetic Algorithm Application:

The setup of the model of informed and uninformed agents is the same as the above described Bray' setup, except, of course, for the learning scheme that agents use. Instead of least squares, uninformed traders use GA to obtain estimates of the intercept and the slope.

The representation and the interpretation of GA part differs from those used in Competitive Firms Model and OLG models where each individual's decision rule was represented by one string only. Here each uninformed agent has a whole population of strings (the number of strings in a population is 30). The first half of a bit string is decoded as value of the estimate of the intercept, a and the second as value of the slope, b . Thus, each agent has a whole population of different estimates and in each time period, before the trading is to take place, decides what his demand for the risky asset will be, using estimates of a string that has the highest fitness value. (The computation of fitness values is explained below.)

In each generation, before GA part takes place, there is a trading period that lasts for t trading iterations.

In every trading iteration t , each uninformed agent chooses a string that has the lowest estimate of a mean square error of the past predictions to determine his estimates of the intercept and of the slope for that trading iteration.

The price of the risky asset in each trading period is determined in the following way:

$$N_i\theta_i[E(r_t|X_t^s, P_t) - P_t] + N_u\theta_u[a_{t_1} + b_{t-1}P_t - P_t] = X_t^s$$

$$N_i\theta_i E(r_t|S_t) - N_i\theta_i P_t + \theta_u \sum_{j=1}^{N_u} a_{t-1}^j + \theta_u P_t (\sum_{j=1}^{N_u} b_{t-1} - 1) = S_t$$

$$P_{t,n} = \frac{S_t - N_i \theta_i E(r_t | S_t) - \theta_u \sum_{j=1}^{N_u} a_{t-1,n}^j}{\theta_u (\sum_{j=1}^{N_u} b_{t-1,n}^j) - N_i \theta_i}$$

where $a_{t-1,n}^j$ is the decoded value of the first half of a string chosen by an uninformed agent j and $b_{t-1,n}^j$ is the decoded value of the second half of a chosen string; for $t = 1$ in the first generation, each agent chooses a string randomly; for all $t > 1$ a string that has the lowest estimate of mean square error is chosen.

After the price of a trading iteration has been determined, the uninformed agents get to know what the actual return on the asset was and then the mean square error of each string in each population is computed and estimates of mean square errors updated according to the formula:

$$m_t^j = m_{t-1}^j + \frac{1}{t} [(\epsilon_t^j)^2 - m_{t-1}^j] \quad (4)$$

$$\epsilon_t^j = r_t - E_{t-1}^j(r_t) \quad (5)$$

Using average mean square errors, fitness values of all strings are calculated. The fitness of k^{th} string in the population of j^{th} agent is given by:

$$\mu_t^{k,j} = \frac{1}{m_t^{k,j}}$$

where superscripts k and j stand for a k^{th} string in the population of j^{th} agent.

After the completion of a trading period, the genetic operators, reproduction, crossover and mutation, take place. The newly reproduced population of chromosomes is then tested against the data of the previous trading period. Using the prices and the returns on the asset, their estimates mean squares errors are computed and the fitness values determined. These fitness values are compared to those of their parents. If a child has a fitness higher than the fitness of either of its parents, it enters into the population of a new generation replacing a parent with a lower fitness. If a parent has a higher fitness value than its child, it remains in the population. After members of the new generation have been determined, a new trading period starts.

We can think of this process as if before the beginning of each trading period, uninformed agents generated, using GA operators, new set of estimates of the slope and of the intercept and tested this new set against the data from the last trading period. Each newly generated string that has a lower average mean square error compared to its parent enters the set of strings that will be used in the next, actual trading period. Old strings that have lower average mean square error than their children remain in the set that is used in the next trading cycle.

(I tried the simulations of the model using pure genetics, i.e., letting children generated through reproduction, crossover and mutation enter the new generation and participate in the new trading period without previous comparison of their fitness values to those of their parents, but the algorithm was not converging. Simulations lasted for 1000 generations.) I ran 26 simulations for different parameter values.

GA converged to the rational expectations equilibrium, i.e. average values of estimates of the intercept and the slope converged to rational expectations and mean value of the actual price converged to the rational expectations price (or very close to it) that would prevail in the market if all agents were informed.

The results of the simulations that have parameter values for which least squares algorithm is unstable are presented in table 4 (simulations 13-24).

Mean values of both rational expectations prices and actual prices were obtained by averaging over iterations of a trading period. Values of estimates of the intercept and the slope represent averages over all estimates used by all agents and over trading period iterations.

5. FUTURE RESEARCH

Further modifications and extensions of these models as well as new applications of learning by GAs include:

1. Extensions for the competitive firms model which are:

- introduction of a stochastic component into the demand for firms' output;
- design of a model in which every firm has its own population of strings, study of its dynamics and comparison of its dynamics to the model described in section 1 (a probability of selection of a string whose decoded value is interpreted as a firm's production decision for a particular period depends on frequencies of occurrences of that string in a population).

2. The application of GA with diploidy and dominance operator to the OLG environment

In artificial genetic search, diploid is a double-stranded chromosome that carries one or more pairs of strings, each containing information for the same functions, but with different function values. Diploidy provides a mechanism for remembering alleles and allele combinations that were previously useful. Dominance operator shields these remembered alleles from harmful selection in a currently hostile environment. (We may think of this as a protection of a distributed, long-term memory against rapid destruction.)

With the addition of dominance operator to GA, selection of chromosomes whose copies enter into the mating pool is performed the same way as in the basic GA, but the crossover, that follows reproduction, is performed within a single chromosome. Two strings that belong to the same chromosome are crossed over in order to obtain two "gametes", strings that participate in the formation of a new population. Mating of chromosomes is performed randomly. Once chromosomes' pairs are determined, gametes that belong to chosen chromosomes are randomly mated to form a new pair of chromosomes. After creation of a new population of double-stranded chromosomes, dominance operator is applied: Alleles that are at the same loci in two strings, which belong to a chromosome, are paired and at each locus, a single allele is chosen to be expressed in a string that will serve as a basis for the evaluation of chromosome's fitness. Alleles can have dominant or recessive values. A dominant allele takes precedence over the alternative, recessive allele at the same locus and is the one that is expressed. A recessive allele is expressed only when coupled with another recessive allele. A result of dominance mapping is a string, one for each chromosome, that is decoded and whose fitness value represents a fitness value of a chromosome.

Dominance and diploidy enable the addition of a kind of "experience" or "memory" to the algorithm. Some experience from past, different conditions is kept and may become useful again if the same conditions were to repeat. GA should adjust faster than standard algorithms to a change if the conditions related to the change occurred at some point in time.

The addition of dominance operator does not change the possible interpretation of the GA process. Reproduction, crossover and mutation can still be interpreted the same way, as imitation of successful rules and generation of new ideas, except that, in this case, it includes previous "knowledge" of well performing strings that is copied and, through recombination and mutation, used to create better rules.

Since the redundant memory permits multiple solutions to the same type of problem to be carried along with only one particular solution expressed, GA with the dominance operator provides very convenient and interesting setup for studying learning under varying monetary policy. I intend to study the behavior of the system, the characteristics of its dynamics, how fast it goes from one rational expectations equilibrium to the other after a policy change has occurred and all the other issues relevant to the analysis of the convergence properties.

3. The application of GA to the OLG model in which sunspot equilibria are possible

This model is essentially the version of Woodford's (1989) model, with GA learning scheme.

Agents observe a random variable s_t (the "sunspot" variable) which takes a finite number of values $\{1, \dots, m\}$ and follows a Markov process with transition probabilities $\Pi_{i,j} > 0$ for $i, j = 1, \dots, m$. The sunspot variable has no effect upon the economy except through agents' expectations that may be conditioned upon it.⁴

Each string is a set, which contains m elements, of beliefs of an individual agent. An agent makes a decision about his action on the basis of his beliefs which are determined in the following manner: if the sunspot variable takes i^{th} value, i^{th} part of an agent's string is decoded as agent's belief about his action in that period, for $i = 1 \dots m$.

Woodford(1989) has shown that there exist sunspot equilibria that are stable under adaptive learning rule and it is possible for the monetary steady state to be unstable for arbitrary m .

Although it is rational for agents to make their decisions dependant on the values of a sunspot variable once they start to believe in it, the possibility that the sunspot equilibria exist requires the assumption of preexisting coordination of beliefs, but it is not clear how this coordination occurs. A setup like the one outlined above would provide a way to model explicitly the evolution of agents' beliefs: whether agents ever begin to believe in the sunspot variable and, in that way, create conditions under which such a belief is rational and whether and how these beliefs can become coordinated to result in the sunspot equilibrium.

4. Further extensions of the model of asset trading by informed and uninformed agents

One of the extensions is a design of a model in which, instead of the selection criterion presented in section 4 (a string with the lowest mean square error is the one whose value is decoded as a belief of an uninformed agent), the probability of selection will depend on frequencies of occurrences in a population.

I plan to have two variants of this extension that differ in the way in which fitness values will be computed. One will have a fitness of a string calculated in the same way as described in section 4, i.e. as an inverse of the mean square error, while in the other a fitness of a string will directly depend on the utility which is a result of a decision taken by an agent, on the basis of beliefs represented by that string. This will allow me to compare the performance of two different kinds of algorithms, one that contains a forecasting rule

⁴ See Woodford (1989) for the full description of the model's setup.

and the other whose only guidance through the search space is utility that it brings to an agent.

5. Comparison of GA behavior with experimental data of OLG economies

I have already started work on this topic by running GA simulations with parameter values and initial values of variables that are equal to the values obtained in the experiments. As has been indicated in section 3, I ran simulations for Lim, Prescott and Summers (1987) OLG economies and for Marimon and Summers (1988) OLG economies.

For the constant money supply OLG model, the algorithm converged to a stationary, monetary equilibrium. The difficulty in comparing GA performance to the experimental results was that experiments for the constant money supply were run for a very short period of time (7 periods). In the case of the seignorage financed deficit, GA simulations show, relatively early in simulations, a bias towards constant consumption which is also present in Marimon and Summers (1987) experimental data. I intend to analyze further GA populations that result in the bias and to try to give a possible explanation for their behavior. (After this initial bias they do converge to a low inflation stationary equilibria.) Marimon and Summers have recently run new experiments within similar setting. I am planning to use that data for further work on this part of my research.

6. Application of classifier systems to Bewley-Townsend model of money ⁵

This is a version of Bewley-Townsend model in which two types of infinitely-lived agents, with oscillating and negatively correlated endowments and borrowing constraints, can only use fiat money to smooth their consumption shares and in which agents use classifier systems to make their savings (consumption) decisions.

A classifier system receives a message, either from its environment or from some other classifier system, which is compared to the condition part of all classifiers that are classifier system's members. Those classifiers whose condition matches a message compete against each other for the right to post a new message. They compete by submitting their bids and the classifier that has the highest bid is a winner and its action part is interpreted as a message sent from the system. This message may either go to another classifier system or may be interpreted as a system's response to a stimulus from its environment. A bid that depends on a classifier's strength (the measure of its performance) is subtracted from a strength of a classifier that won a competition and is rewarded to a classifier (added to its strength) that wins another cycle of competition, this time among classifiers of a system to which a message from the first classifier system was sent.

In this version of Bewley-Townsend model, each agent (or each type of agent) uses three types of classifier systems. The first is a price forecasting classifier system that sends its prediction to the second classifier system. The second system, besides this prediction, gets a message about money balances and endowment of an agent and it posts a message about whether to save or spend in a particular time period to the third classifier system. The third classifier system consists of two subpopulations of classifiers which are alternatively used depending on whether an agent decided to save or dissave in that period. This

⁵ I have already started working on this project together with R. Marimon and T. Sargent

classifier decides on a fraction of consumption good (money balances) that will be saved (spent).

Once each agent makes his savings or dissavings decision for a particular time period, the market price of a consumption good for that period is determined. Utility that each agent obtains (which depends on his own decision and on the market price) serves as a basis for evaluation of classifier systems' rewards. The winner of the competition among classifiers in the third system gets as its reward utility that agent acquired (its strength is increased by that amount) which is propagated back into the system by credit assignment briefly outlined above. This way, all classifiers that contributed to an agent's decision get rewards that depend on how good a decision was.

Classifier systems are initialized by randomly generated rules. GA is performed on rules' condition and action parts every prespecified number of periods.

A starting point of the research is a model with a constant money supply. The whole research project includes studying of more complex versions of this model, such as random endowments, alternative monetary policies etc.

TABLE 1

MODEL OF COMPETITIVE FIRMS THAT LEARN TO PREDICT THE CORRECT MARKET PRICE
(all simulations were run for 200 generations)

	Model 1	Model 2	Model 3	Model 4	Model 5
A	100	10	100	7	1,000
B	0.02	0.03	0.02	0.003	0.02
X (intercept)	3	2	1	2	200
Y (slope)	1	1	1	1	1
size of the population	20	20	30	30	30
chromosome length	20	20	30	20	30
convergence achieved after generation	45	19	190	38	47
price after convergence*	72.28516	7	62.875	6.587	700
rational expectations** quant. for above price	69.2856	5	61.875	4.587	500
actual quant. after convergence	69.28711	5	61.875	4.587	500
profit (fitness) after convergence	2400.216	12.5	957.031	10.520	124,999.999

$$*p_t = A - B \cdot \sum q^i \quad (q^i - \text{individual quantity})$$

$$**Q_t = E_{t-1}P_t - X$$

$$E_{t-1}P_t = p_t$$

TABLE 2

Overlapping generations model with constant money supply

Model	I	II	III	IV
e1	150	120	100	7
e2	10	20	90	1
M ^s	1,000	500	1,000	1,000
P ^s	14.286	10	200	333.33
Cs1	80	70	95	4
Cs2	80	70	95	4
Pa	14.286	10	196.923	333.33
Ca1	80	70	94.92187	4
Ca1	80	70	95.07813	4
Ua	6,400	4,900	9,024.9939	16

- e1, e2 - endowment in the first and the second period (young and old) respectively
- M^s - nominal per capita money supply
- P^s - price level in the stationary, rational expectations equilibrium
- Cs1, Cs2 - consumption in the first and the second period, respectively, in the stationary equilibrium
- pa - actual price level after the convergence of GA
- Ca1, Ca2 - consumption in the first and the second period, respectively, after the convergence of GA
- Ua=Ca1 x Ca2 - utility (string fitness) after the convergence of GA

TABLE 3

OLG Model with Constant Deficit Financed through Seignorage

Model	I	II	III	IV
e^1	150	150	100	1
e^2	30	10	90	0.9
d	15	0.5	0.007695	0.000972
$b=e^1/e^2$	5	15	1.111	1.111
$a=b+1$				
$-(2d/e^2)$	5	15.9	2.11094	2.1089511
Π_1^s	1.382	1.007	1.00156	1.026
Π_2^s	3.618	14.326	1.10938	1.083
Π^i	1.560	1.326		
Π^a	1.382	1.007	1.00156	1.026
ca^1/ca^2	1.382	1.007	1.00156	1.026

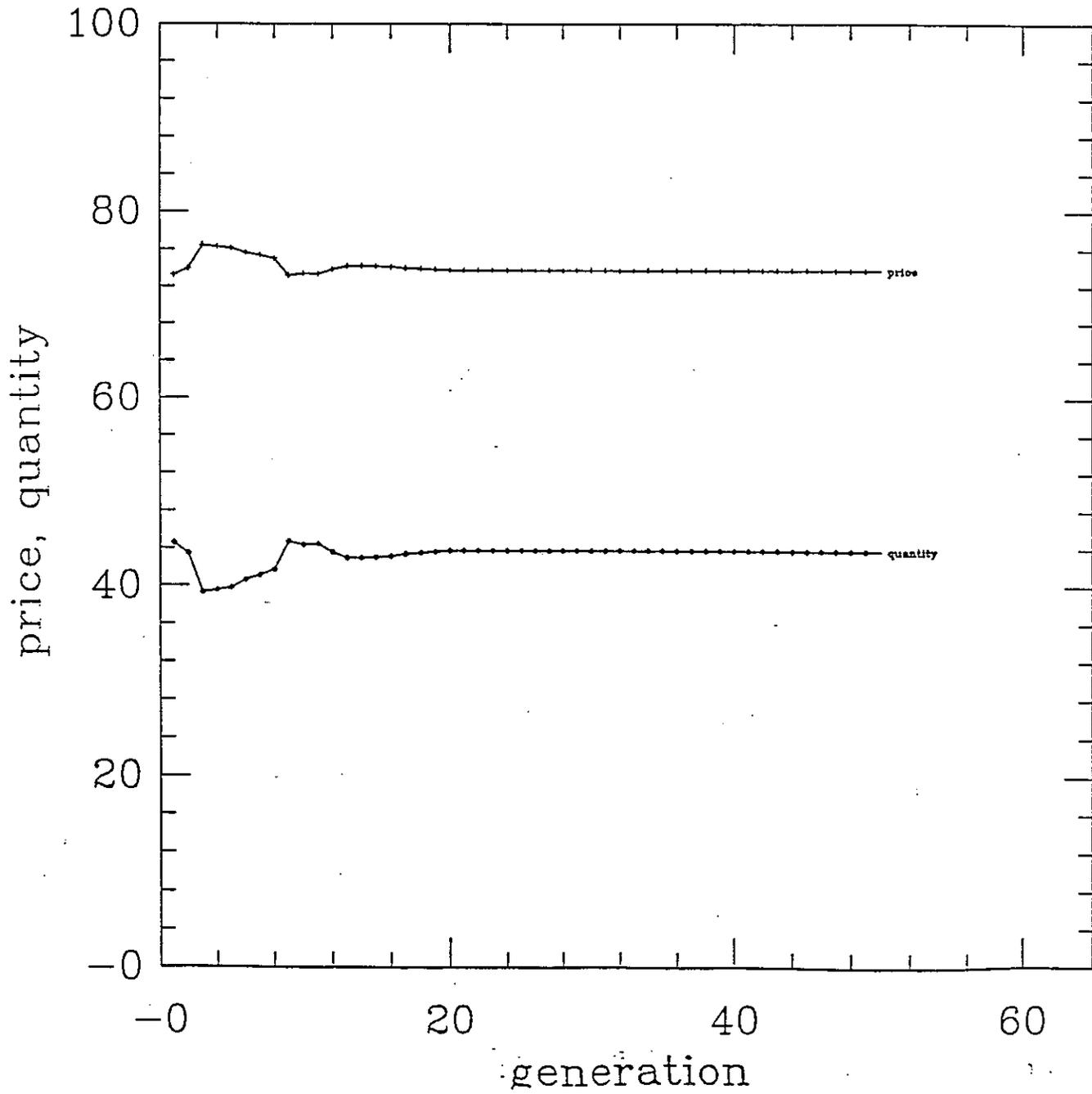
- e^1, e^2 - endowment in the first and the second period (young and old) respectively
- d - per capita, constant, real deficit
- inflation rates for two stationary, rational expectations equilibria
- Π^i - initial inflation rate in a simulation
- Π^a - actual inflation rate after the convergence of GA
- ca^1/ca^2 - actual consumption ratio after the convergence of GA

Model of Informed and Uninformed Agents

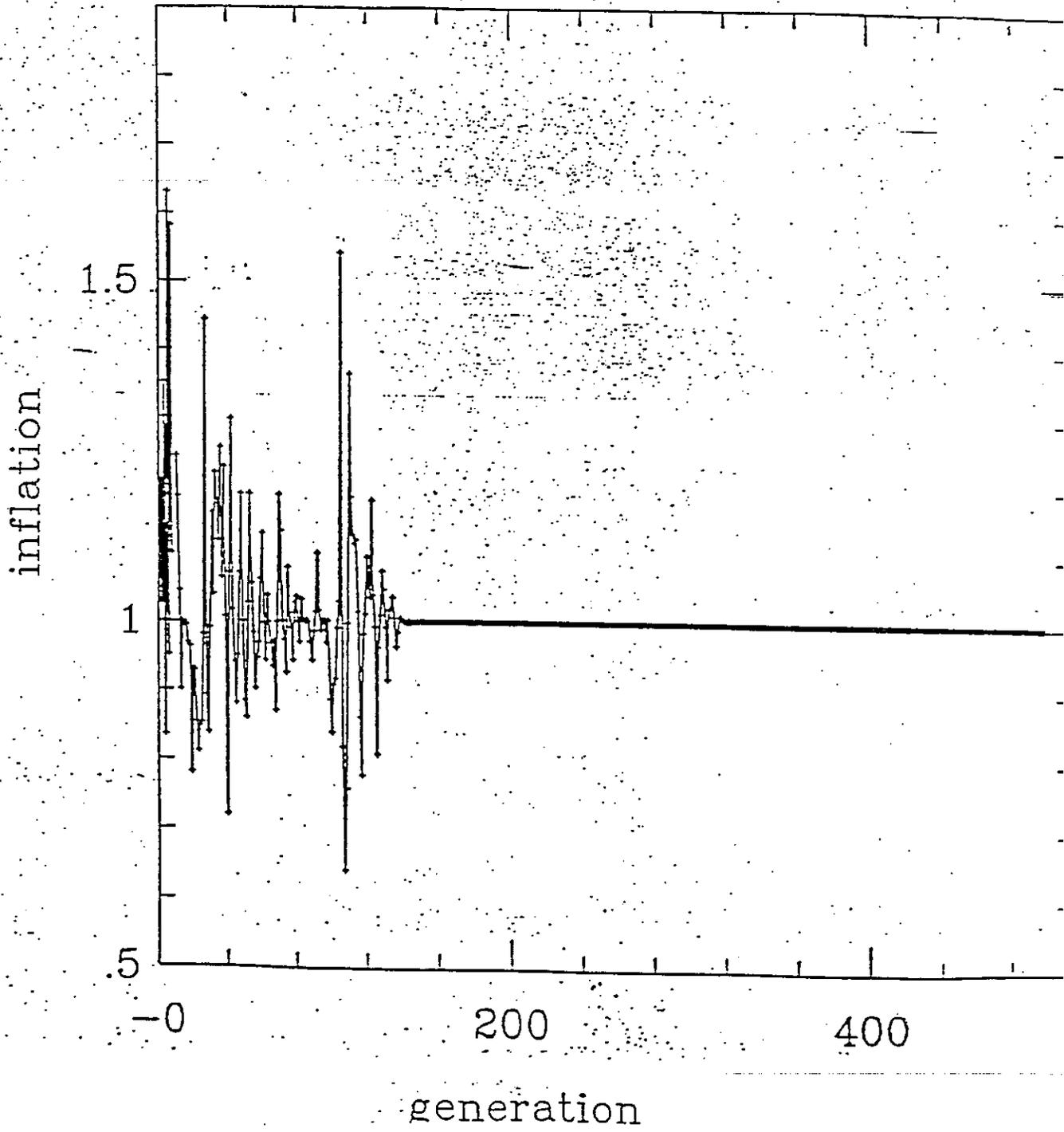
	13	14	15	16	17	18	19	20	21	22	23	24
number of uninformed	40	40	40	40	40	40	40	40	40	40	40	40
number of informed	0	370	500	500	100	20	0	10	62	62	50	50
number of generations	300	300	300	300	300	300	300	300	300	300	300	300
number of trading interactions	50	50	50	50	50	50	50	50	50	50	50	50
r	100	100	100	160	100	100	160	160	100	160	100	160
S	3,000	30,300	40,000	40,000	12,000	5,140	3,430	4,230	7,650	7,650	6,750	6,750
average rat. expectations price	25	26.10	25.93	85.93	14.29	14.34	14.25	75.40	25	85	25	85
average GA price after convergence	25.04	26.10	25.93	85.74	14.30	14.36	14.29	73.55	25.01	84.01	25.02	83.87
p	0.0158	0.0158	0.0158	0.0158	0.0354	0.0354	0.0354	0.0354	0.0158	0.0158	0.0158	0.0158

TABLE 4

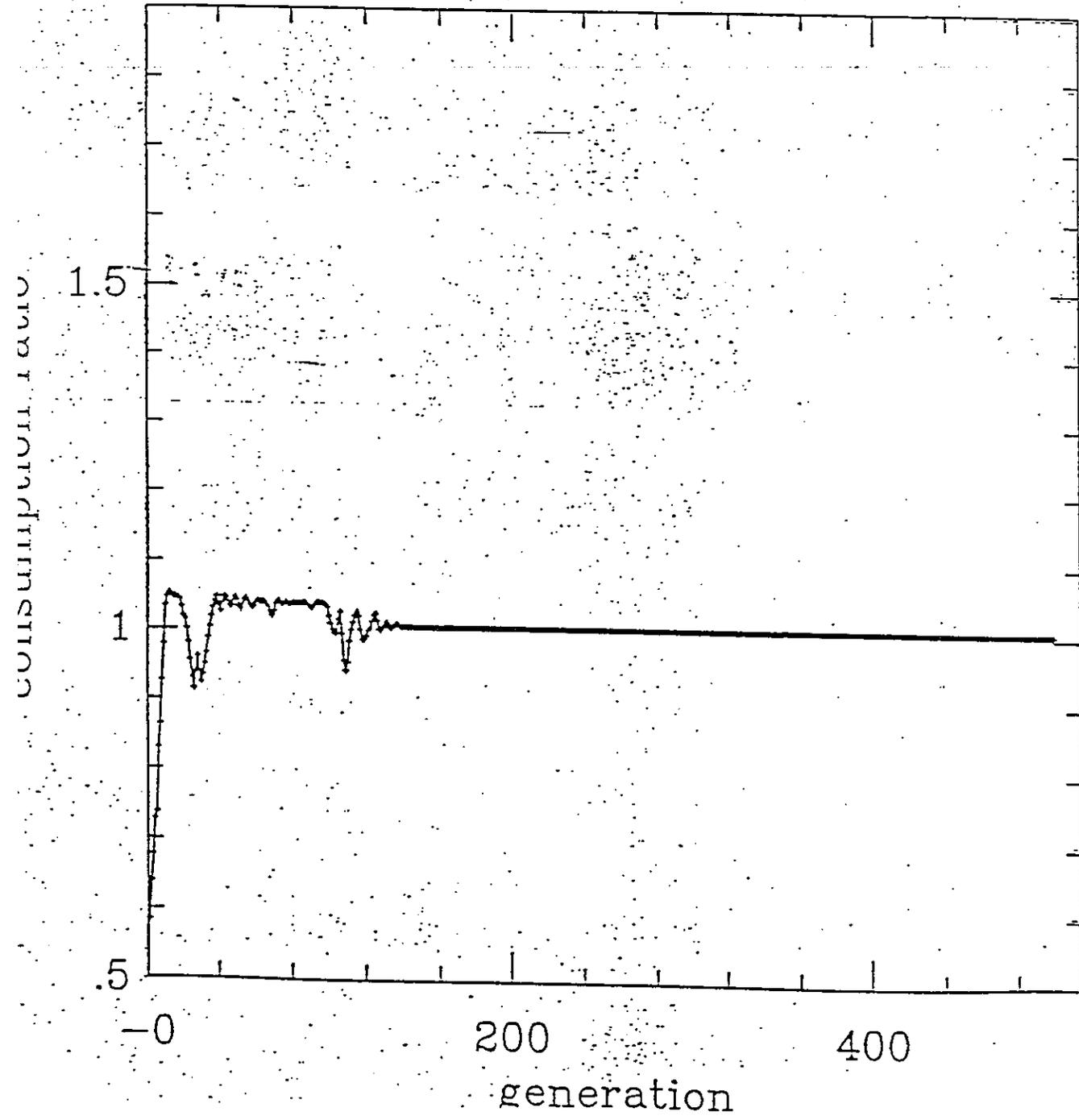
Competitive firms model (outputin.dat)



OLG model (outputin.dat)



OLG model (output.dat)



REFERENCES

Axelrod, R., "The simulation of genetics and evolution", presented at a Conference on Evolutionary Theory in Biology and Economics, University of Bielefeld, Federal Republic of Germany, 1985

Axelrod, R., "The evolution of strategies in the iterated prisoner's" in Davis L. (Ed.), "Genetic Algorithms and Simulated Annealing" (p.32-41), London: Pitman, 1987

Bagley, J.D., "The behavior of adaptive systems which employ genetic and correlation algorithms" (Doctoral dissertation, University of Michigan), Dissertation Abstracts International, 28(12), 5106B, (University Microfilms No. 68-7556), 1967

Bray, M.M., "Convergence to rational expectations equilibrium" in Frydman, R. and Phelps, E.S. (Ed.). "Individual forecasting and aggregate outcomes", Cambridge University Press, 1983

Bray, M.M., "Learning, Estimation and the Stability of Rational Expectations", Journal of Economic Theory 26 (1982), p.318-339

Bewley, T. "The Optimum Quantity of Money", in Karaken J.H. and N. Wallace, (Ed)., "Models of Monetary Economics", Federal Reserve Bank of Minneapolis, Minneapolis, 1980, p. 49-82.

Caviccio, D.J., "Adaptive search using simulated evolution", Doctoral dissertation, University of Michigan, 1970

De Jong, K.A., "An analysis of the behavior of a class of genetic adaptive systems", (Doctoral dissertation, University of Michigan), Dissertation Abstracts 36(10), 5140B, (University Microfilms No. 76-9381), 1975

De Jong, K.A., "A Genetic-based global function optimization technique" (Technical Report No. 80-2), University of Pittsburgh, Department of Computer Science, 1980

Edelman, G.M., "Neural Darwinism", Basic Books, Inc., 1987

Fitzpatrick, J.M., Grefenstette, J.J. and Van Gucht, D., "Image registration by genetic search", Proceedings of IEEE Southeast Conference, 460-464, 1984

Frydman, R., "Towards an Understanding of Market Processes," *American Economic Review*, 72 No.4, 652-668 (September 1982)

Goldberg, D.E., "Computer-aided pipeline operation using genetic algorithms and rule learning", Paper presented at the 1984 API Pipeline Cybernetics Symposium, Houston, 1984

Goldberg, D.E. and Samtani, M.P. "Engineering optimization via genetic algorithm", *Proceedings of the Ninth Conference on Electronic Computation*, 471-482

Goldberg, D.E., "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley Publishing Company, 1989

Grossman, S.J. and Stiglitz, J.E., "Information and competitive price systems", *American Economic Review* 66, 1976, p.246-253

Holland, J.H., "Robust algorithms for adaptation set in a general formal framework", *Proceedings of the IEEE Symposium on Adaptive Processes - Decision and Control*, XVII, 5.1-5.5, 1970a

Holland, J.H., "Processing and processors for schemata", in Jacks, E.L. (Ed.), "Associative information processing" (p. 127-146), American Elsevier, New York, 1970b

Holland, J.H., "Adaptation in Natural and Artificial Systems", The University of Michigan Press, 1975

Holland, J.H., Holyoak, K.J., Nisbett, R.E. and Thagard, P.R., "Induction - Processes of Inference, Learning and Discovery", The MIT Press, 1986

Lim, S.S., Prescott, E.C. and Sunder, S., "Stationary Solution to the Overlapping Generations Model of Fiat Money: Experimental Evidence", mimeo, March 1988

Lucas, R.E., Jr., "Adaptive Behavior and Economic Theory", *Journal of Business*, 1986, vol.59, no.4, pt. 2

Marcet, A. and Sargent, T.J., "Convergence of Least Squares Learning Mechanisms in Self Referential Linear Stochastic Models", *Journal of Economic Theory*, vol.48, No.2, August 1989, p. 337-368

Marcet, A. and Sargent, T.J., "Least Squares Learning and the Dynamics of Hyperinflation", "Economic Complexity: Chaos, Sunspots, Bubbles and Nonlinearity", (Ed.), Barnett, W.A., Geweke J. and Shell K." Cambridge University Press, 1989

Marcet, A. and Sargent, T.J., "Convergence of Least Squares Learning in Environments with Hidden State Variables and Private Information", mimeo, December 1987

Marimon, R. and Sunder, S., "Rational Expectations vs. Adaptive Behavior in a

Hyperinflationary world: Experimental Evidence", June 1988

Marimon, R., McGrattan E. and Sargent T.J., "Money as a Medium of Exchange in an Economy with Artificially Intelligent Agents", 1989, forthcoming in *Journal of Economic Dynamics and Control*

Miller, J.H., "A Genetic Model of Adaptive Economic Behavior", University of Michigan, mimeo, 1986

Miller, J.H., "The Evolution of Automata in the Repeated Prisoner's Dilemma", University of Michigan, mimeo, 1987

Muth, J.F., "Rational Expectations and the Theory of Price Movements", *Econometrica* 1961, vol 29, no.6

Rumelhart, D.E., McClelland, J.L. and the PDP Research Group, "Parallel Distributed Processing", The MIT Press, 1987

Samuelson, P.A. 1958, "An Exact Consumption-loan Model of Interest with or without the Social Contrivance of Money", *Journal of Political Economy* 66, (December) p. 467-482

Sargent, T.J., "Dynamic Macroeconomic Theory", Harvard University Press, 1987

Townsend, R. "Models of Money with Spatially Separated Agents", in Karaken, J.H. and Wallace, N. (Ed.), "Models of Monetary Economics", Federal Reserve Bank of Minneapolis, Minneapolis

Wallace, N. "The Overallping Generations Model of Fiat Money", in Karaken, J.H. and Wallace, N. (Ed.), "Models of Monetary Economics", Federal Reserve Bank of Minneapolis, Minneapolis, 1980, p. 49-82

Woodford, M. "Learning to Believe in Sunspots", mimeo, University of Chicago, February 1989, forthcoming in *Econometrica*

Working Papers, Preprints, and Reprints in the SFI Series

- 89-001 "A Double Auction Market for Computerized Traders"
John Rust, Richard Palmer, and John H. Miller
- 89-002 "Communication, Computability and Common Interest Games"
Luca Anderlini
- 89-003 "The Coevolution of Automata in the Repeated Prisoner's Dilemma"
John H. Miller
- 89-004 "Money as a Medium of Exchange in an Economy with Artificially Intelligent Agents"
Ramon Marimon, Ellen-McGrattan, and Thomas J. Sargent
- 89-005 "The Dynamical Behavior of Classifier Systems"
John H. Miller and Stephanie Forrest
- 89-006 "Nonlinearities in Economic Dynamics"
José A. Scheinkman
- 89-007 "'Silicon Valley' Locational Clusters: When Do Increasing Returns Imply Monopoly?"
W. Brian Arthur
- 89-008 "Mutual Information Functions of Natural Language Texts"
Wentian Li
- 90-001 "Learning by Genetic Algorithms in Economic Environments"
Jasmina Arifovic
- 90-002 "Optimal Detection of Nonlocal Quantum Information"
Asher Peres and William K. Wootters
- 90-003 "Artificial Life: The Coming Evolution"
J. Doyne Farmer and Alletta d'A. Belin
- 90-004 "A Rosetta Stone for Connectionism"
J. Doyne Farmer
- 90-005 "Immune Network Theory"
Alan S. Perelson
- 90-006 "Protein Evolution of Rugged Landscapes"
Catherine A. Macken and Alan S. Perelson
- 90-007 "Evolutionary Walks on Rugged Landscapes"
Catherine A. Macken, Patrick S. Hagan, and Alan S. Perelson

Economic Research Program Titles

- 89-001 "A Double Auction Market for Computerized Traders"
John Rust, Richard Palmer, and John H. Miller
- 89-002 "Communication, Computability and Common Interest Games"
Luca Anderlini
- 89-003 "The Coevolution of Automata in the Repeated Prisoner's Dilemma"
John H. Miller
- 89-004 "Money as a Medium of Exchange in an Economy with Artificially Intelligent Agents"
Ramon Marimon, Ellen McGrattan, and Thomas J. Sargent
- 89-005 "The Dynamical Behavior of Classifier Systems"
John H. Miller and Stephanie Forrest
- 89-006 "Nonlinearities in Economic Dynamics"
José A. Scheinkman
- 89-007 "'Silicon Valley' Locational Clusters: When Do Increasing Returns Imply Monopoly?"
W. Brian Arthur
- 90-001 "Learning by Genetic Algorithms in Economic Environments"
Jasmina Arifovic

