WORKING PAPER SERIES

A Model of Learning and Emulation with Artificial Adaptive Agents

James B. Bullard
John Duffy

Working Paper 1994-014C
http://research.stlouisfed.org/wp/1994/94-014.pdf

PUBLISHED: Journal of Economic Dynamics and Control, February 1998.

FEDERAL RESERVE BANK OF ST. LOUIS
Research Division
411 Locust Street
St. Louis, MO 63102

A MODEL OF LEARNING AND EMULATION WITH ARTIFICIAL ADAPTIVE AGENTS

December, 1996

ABSTRACT

We study adaptive learning behavior in a sequence of n-period endowment overlapping generations economies with fiat currency, where n refers to the number of periods in agents' lifetimes. Agents initially have heterogeneous beliefs and seek to form multi-step-ahead forecasts of future prices using a forecast rule chosen from a vast set of possible forecast rules. Agents take optimal actions given their forecasts of future prices. They learn in every period by creating new forecast rules and by emulating the forecast rules of other agents. Computational experiments with artificial adaptive agents are conducted. These experiments yield three qualitatively different types of outcomes. In one, the initially heterogeneous population of artificial agents learns to coordinate on a low inflation, stationary perfect foresight equilibrium. In another, we observe persistent currency collapse. The third outcome is a lack of coordination within the allotted time frame. One possible outcome, a stationary perfect foresight equilibrium with a relatively high inflation rate, is never observed.

KEYWORDS: Learning, genetic algorithms

JEL CLASSIFICATION: D83, C63

James Bullard
Research Officer
Federal Reserve Bank
411 Locust Street
St. Louis, MO 63102
(314) 444-8576--fax (314) 444-8731
bullard@stls.frb.org

John Duffy
Department of Economics
University of Pittsburgh
Pittsburgh, PA 15260
(412) 648-1733--fax (412) 648-1793
jduffy+@pitt.edu

We thank John Miller, an associate editor and two anonymous referees for helful comments and suggestions. We have also benefitted from the comments of participants at the February 1994 NBER conference on macroeconomic complementarities, the 1994 IFAC meeting on Computational Methods in Economics, the Summer 1994 Econometric Society meetings, and the Summer 1994 SEDC meetings.

1 Introduction

1.1 Overview

Macroeconomic research is frequently conducted using general equilibrium models in which agents must forecast endogenous variables many periods ahead in order to make consumption and savings decisions. Examples in this class include Ramsey–type growth models and multi–period overlapping generations models. Much of the recent literature on adaptive learning behavior, however, focuses on models where agents must plan only one period ahead.[1] In this paper, we examine adaptive learning behavior in an environment where agents must form plans many periods ahead. Our analysis involves computational experiments with artificial adaptive agents who populate a sequence of n–period endowment overlapping generations economies, where n refers to the number of periods in agents' lifetimes. We assume that agents within and across generations initially have heterogeneous beliefs about the future path of prices and rates of return. The questions we seek to address are (1) how do initially heterogeneous agents learn over time to forecast the future correctly in general equilibrium environments, and (2) can such a learning process be used to isolate a particular stationary equilibrium solution in a model with multiple stationary equilibria.

In order to address the first question, we develop a model that makes use of Holland's (1975) *genetic algorithm*, a stochastic directed search algorithm that is based on principles of population genetics.[2] The genetic algorithm processes a finite population of objects each of which can be thought of as representing an individual agent. In our case, the objects represent agents' forecast rules. The objects are encoded as strings in a finite (usually binary) alphabet. Strings can be decoded and evaluated according to some fitness criterion. In every iteration, the genetic algorithm replaces the fixed population of strings by choosing among existing strings in a manner that is consistent with the principle of natural selection; strings with high fitness values are more likely to be chosen than strings with low fitness values. The chosen strings also undergo genetic operations of recombination and mutation which serve to advance the search for higher fitness. When this process is applied repeatedly, eventually only highly fit strings remain in the population, and often a single string will come to dominate the population. Holland (1975) has argued that genetic algorithms optimize on the trade–off between searching for new solutions and utilizing information discovered in the past.[3]

[1] For a survey of the literature, see Sargent (1993).

[2] See Goldberg (1989) or Michalewicz (1994) for an introduction to genetic algorithms.

[3] Holland's "schema theorem" is proved by an analogy with the two–armed bandit problem and is found in Holland (1975, pp. 75–88). Goldberg (1989, pp. 28–33) provides a less technical description of Holland's result.

The genetic algorithm is a natural choice for modeling the manner in which a heterogeneous population of agents might go about learning to forecast future prices. However the genetic algorithm as it is typically implemented is not well–suited for handling an important aspect of our relatively standard general equilibrium problem. In particular, the standard genetic algorithm does not allow objects that are part of the system for more than one iteration to be updated. Agents who are in the middle of life in our model may want to change their forecast rule over the remaining periods of life instead of maintaining their original forecast rule (the one they adopted in the first period of life). We therefore augment the genetic algorithm with an additional procedure that allows each agent in our model to emulate the behavior of other agents. Even the most casual observation suggests that emulation of other agents' is a large part of adaptive learning behavior. We show how the desire to emulate other agents' forecast rules arises naturally in the n–period overlapping generations environment that we consider.

We interpret our learning and emulation model as describing a process by which ideas are exchanged in a large, competitive economy with heterogeneous beliefs. We view the genetic algorithm as capturing a trial–and–error learning process through which successful ideas (forecast rules) are propagated while new ideas continually emerge. These new ideas come from two sources: (1) the newly created forecast rules of "newborn" agents, and (2) emulation. Our interpretation of the model is intended to de–emphasize the obvious biological undertones of both the overlapping generations model and the genetic algorithm.

In answer to the second question that we posed, as to whether such a learning process can be used to isolate a particular equilibrium in an environment with multiple equilibria, we report the results of computational experiments using many different parameterizations of our model of learning and emulation. In each of these experiments, the initial population of forecast rules is randomly generated. We examine what happens to this population of forecast rules over many iterations using our genetic–algorithm–based model of learning and emulation.

1.2 Summary of findings and conclusions

We find that three different outcomes are possible after we simulate our system for a long period of time. First, we often observe coordination on a stationary equilibrium with a relatively low inflation rate. There are two stationary perfect foresight equilibria in the model, one with a relatively low inflation rate and one with a relatively high inflation rate, but coordination on the stationary equilibrium with the relatively high inflation rate is never observed. This first result is consistent with the findings of many earlier analyses of adaptive learning behavior in two period overlapping generations economies.[4]

Second, we often observe persistent currency collapse. It can happen that some agents' forecast rules yield predictions of very high inflation rates. These agents will then decide to hold little or no currency, and by itself (that is, holding the decisions of the other agents constant) this effect will tend to push next period's inflation higher. Higher inflation can then validate the forecasts of higher inflation, so that the forecast rules of this group turn out to be successful (they achieve high levels of fitness). This can lead to a propagation of forecast rules that predict high inflation, which can in turn cause actual inflation to remain high. In this way, a self-fulfilling hyperinflation can be achieved and sustained. We interpret this outcome as a *currency collapse*, as fiat money would cease to have value were it not for some restrictions that we place on agent's portfolio decisions. In our computational experiments, this currency collapse outcome appears to become more frequent as the number of periods that agents live increases and as the level of the government deficit increases.[5]

The third result is that the population of adaptive agents sometimes does not achieve coordination on *any* stationary outcome within the allotted time. We can only speculate that these nonconvergent cases might eventually converge to a perfect foresight equilibrium outcome or to a self-fulfilling hyperinflation if the simulations were allowed to continue for a longer period.

Our general conclusions are as follows. First, our algorithm can be given an appealing economic interpretation as a model of trial-and-error learning through which older, successful ideas are retained and propagated while some new ideas are developed and exchanged among agents with heterogeneous beliefs and planning horizons, all of whom are participants in a large competitive economy. Second, our model of learning and emulation is a useful search algorithm that may enable researchers to select from among multiple stationary equilibrium outcomes in the highly nonlinear and multi-period environments that are increasingly being studied by macroeconomists. Third, our computational experiments suggest that currency collapse is a distinct possibility under our trial-and-error learning scheme, even when the government is following an otherwise sustainable deficit policy. Of particular interest is our finding that forecasts may interact with the general equilibrium price level dynamics in such a way as to generate a self-fulfilling hyperinflation.

[4]See, for instance, Marcet and Sargent (1989), Woodford (1990), Marimon and Sunder (1993), and Arifovic (1995).

[5]Evans and Ramey (1995, 1996) can also generate currency collapse outcomes under the assumption that it is costly for agents to form expectations.

1.3 Recent related literature

Computational experiments with adaptive algorithms (such as genetic algorithms) in economic settings have been suggested by Holland and Miller (1991) and Sargent (1993), and have been performed by Andreoni and Miller (1995), Arifovic (1994, 1995, 1996), Arthur (1994), Axelrod (1987), Binmore and Samuelson (1992), Ho (1996), Marimon, McGratten and Sargent (1990), Miller (1996) and Rust, Miller, and Palmer (1994) among others. These applications have generally not been in competitive general equilibrium environments like we have in mind. An important exception is the work of Arifovic (1995, 1996), who studies genetic algorithm learning in two period overlapping generations economies with government issued fiat currency. Arifovic (1995), in particular, reports that the genetic algorithm learning model converges to a low inflation, stationary perfect foresight equilibrium of the two period overlapping generations model even in cases where Marcet and Sargent (1989) found that the model with least squares learning failed to converge. Later in this paper we argue that Arifovic's (1995) non–overlapping information structure is not very promising when agents' lifetimes are extended to n periods. This observation motivates our development of a modified genetic algorithm learning model in which "middle–aged" agents can choose to emulate one another.

The rest of the paper is organized as follows. Section two presents the n–period overlapping generations economy that we study. Section three provides the details of our genetic-algorithm-based model of learning and emulation, and section four reports the results from our computational experiments. Section five concludes with a summary of our findings.

2 An n–period overlapping generations model

We consider a sequence of n–period endowment overlapping generations economies where n represents the number of periods in an agent's lifetime. Time t is discrete and may take on integer values from the infinite past to the infinite future. At any time t, there are n generations of agents alive, with each generation having a different birth date corresponding to times $t-n+1$, $t-n+2$, ..., $t-1$, t. The number of agents in each generation is finite and there is no population growth. For simplicity, we shall assume in this section that there is a single representative agent alive in every generation. We shall also assume for now that each of these agents has perfect foresight knowledge of future prices and rates of return. This last assumption allows us to characterize the stationary equilibria of the model. We will relax both of these assumptions later when we study adaptive learning behavior in this environment. All agents, regardless of the period in which they were born, receive the same lifetime endowment profile for the single, perishable consumption good in the n periods of life. This endowment profile is denoted by $\{w_1, w_2, ..., w_n\}$, and it is further assumed that $w_1 > w_2 > ... > w_n > 0$.

Agents in this economy can save only by holding government issued fiat currency. Our assumption that the endowment profile is strictly decreasing over the n periods of each agent's lifetime implies that agents of all generations will always be able to achieve their optimal consumption and savings decisions by holding fiat currency. We explicitly rule out the possibility of a private consumption loan market, but the elimination of the consumption loan market does not prevent agents from achieving the optimal outcome.[6]

Fiat currency is introduced into this economy by a government that endures forever. This government has control over the fixed level of its real deficit, ξ, which it finances entirely through seignorage:

$$\xi = \frac{H(t) - H(t-1)}{P(t)}. \tag{1}$$

Here $H(t)$ denotes the stock of currency in circulation at time t, and $P(t)$ is the price of the consumption good in terms of fiat money. We assume that all government revenue leaves the economy.

Agents of all generations seek to maximize the same time separable logarithmic utility function. The agent born at time t seeks to maximize

$$U = \sum_{i=0}^{n-1} \ln c_t(t+i)$$

subject to

$$c_t(t) + \sum_{i=1}^{n-1} c_t(t+i) \prod_{j=0}^{i-1} \beta(t+j) \leq w_1 + \sum_{i=1}^{n-1} w_{i+1} \prod_{j=0}^{i-1} \beta(t+j),$$

where $c_t(t+i)$ denotes consumption by the agent born at time t in period $t+i$, and $\beta(t+j)$ denotes the gross inflation rate between dates $t+j$ and $t+j+1$. Under the assumption that all agents have perfect foresight, we have that

$$\beta(t+j) = \frac{P(t+j+1)}{P(t+j)} \quad \forall\, j.$$

[6]We rule out the possibility of consumption loans so as to avoid some difficulties in modelling how a heterogeneous population of agents might meet and agree to borrowing and lending decisions when they lack agreement about future rates of return.

The solution to the maximization problem of the young agent born at time t yields a lifetime consumption and savings plan. The consumption and savings plans of the representative agents born at earlier dates $t-1, t-2, ..., t-n+1$ can be calculated in a similar manner simply by changing the horizon over which the maximization problem is defined. Solving all n of these problems, it is possible to construct the amount of aggregate savings (asset holdings) in the economy at any date t, $S(t)$, by summing together the time t savings amounts of all n individuals alive in the economy at time t. It can be shown that this aggregate savings function is given by:[7]

$$S(t) = \sum_{i=1}^{n-1} w_i + \sum_{i=0}^{n-3} \sum_{j=1}^{n-2-i} w_{i+1} \prod_{k=1}^{j} \beta(t-k)^{-1} - W_1 - \sum_{i=1}^{n-2} \sum_{j=0}^{i} W_{1-i} \prod_{k=1}^{i} \beta(t-k)^{-1},$$

where

$$W_k \equiv \frac{1}{n} \left[w_1 + \sum_{i=1}^{n-1} w_{i+1} \prod_{j=k-1}^{k+i-2} \beta(t+j) \right].$$

Since all savings must be held in the form of fiat currency, the market clearing condition for this economy is given by:

$$S(t) = \frac{H(t)}{P(t)}. \tag{2}$$

Equilibria in this economy are characterized by positive sequences for prices and fiat currency that satisfy equations (1) and (2). Combining equations (1) and (2) one obtains an equilibrium law of motion for prices:

$$P(t) = \left[\frac{S(t-1)}{S(t) - \xi} \right] P(t-1). \tag{3}$$

Stationary perfect foresight equilibria for the general n–period model are solutions to equation (3) under the assumption that the forecast price is equal to the actual price $P(t+j)$ for all j. Bullard (1992) has shown for the case where $\xi = 0$, that there are at most two such stationary values for β for any value of n; this result generalizes to the case studied here of a positive government deficit. To ensure that both of these stationary equilibria exist and involve the use of positive valued fiat currency, we need one further assumption. We require that $\xi \in (0, \bar{\xi})$, where $\bar{\xi}$ is the maximum amount of revenue the government can collect through seignorage.[8] That is, the deficit cannot be so high as to cause nonexistence of equilibria.

[7] See Bullard (1992).

[8] This maximum deficit level, $\bar{\xi}$, can be calculated as follows. In a stationary equilibrium, equation (3) can be rewritten as $\xi = \left(1 - \frac{1}{\beta}\right) S(\beta)$, where $S(\beta)$ is the stationary value of the aggregate savings function given above. To find $\bar{\xi}$ we numerically solved the equation $\partial\xi/\partial\beta = 0$ for β, and then used this value of β to determine $\bar{\xi}$ using equation (3).

In summary, the general n–period model that we consider in this paper has two stationary perfect foresight equilibria in which fiat currency has positive value. We shall denote the stationary equilibrium with the relatively low inflation rate by β_ℓ and we shall denote the stationary equilibrium with the relatively high inflation rate by β_h, where $\beta_\ell < \beta_h$. The general n–period model under perfect foresight is therefore analogous to the more familiar two period endowment overlapping generations economy that has been the focus of previous analyses of adaptive learning behavior. We now turn to our model of how agents learn in the n–period overlapping generations environment when they lack perfect foresight and when there are many heterogeneous agents within each generation.

3 A model of learning and emulation

We begin this section by discussing how a heterogeneous population of agents in a large competitive economy might exchange ideas. We then describe how our heterogeneous population of agents can be represented and how their forecast rules can be evaluated in a manner that is consistent with the application of a genetic algorithm. We also discuss how agents use their forecast rules to generate optimal consumption and savings decisions. Next, we describe how the genetic algorithm can be applied to update the forecast rules of the heterogeneous population of agents and how the genetic algorithm can be augmented to include an emulation procedure. Finally, we provide a summary of the entire algorithm.

3.1 Non–overlapping information structures

The manner in which ideas are exchanged among agents of different generations becomes a critical issue in the general n–period environment that we consider. Arifovic (1995) studies genetic algorithm learning in a two period overlapping generations framework, and imagines that newborn agents inherit ideas from the generation born *two* periods ago—the newborn agent's "grandparents." Generations that overlap with one another do not exchange ideas at all. We refer to this situation as a *non–overlapping information structure.* If we adopted a non–overlapping information structure for our sequence of n–period overlapping generations economies, there would be no interaction between the oldest generation and members of the "middle generations" who overlap with the oldest generation. Agents in middle generations are neither "old" (agents who are in the last period of their productive lives) nor "newborn" (agents who are just beginning the first period of their productive lives). A non-overlapping information structure, applied to our system, would not allow for the possibility that members of these middle generations are also learning from one date to the next. The lack of a learning role for these middle generations is a shortcoming that becomes especially troublesome as the number of periods, n, becomes large. As Sargent (1993, pp. 101-2) notes,

> In the [two period overlapping generations] model, learning occurs between *non-overlapping* generations of grandparents to grandchildren, with the grandchildren adjusting their grandparents' savings choice after observing the consequence of their grandparents' choice... When we extend the horizon beyond two periods, it becomes increasingly inconvenient to model learning in this way because we have to wait longer for the consequences of lifetime savings behavior to be known. [Italics in original.]

The "wait" that Sargent refers to is a burden borne by the middle generations, who cannot determine the relative fitness of their forecast rules until they are in the n^{th} period of life and have knowledge of the last n market prices. It is implausible to think that agents would wait their entire lifetime before revising their forecast rules and consumption and savings plans, especially when a new market price level realization is revealed to them in every period. For this reason, we do not use a non-overlapping information structure in our model.[9] Our main idea is to instead augment the genetic algorithm with an "emulation" procedure.

We now turn to a detailed discussion of our learning and emulation algorithm.

3.2 Representation of a heterogeneous population

The population of agents alive at time t is denoted by $A(t)$. An individual agent at time t is denoted by $A_{ij}(t)$ where the i subscript, $i = t-n+1, ..., t$, refers to the agent's *birth date.* In each generation there are m *members* who are indexed by $j = 1, 2, ..., m$. The total population of agents alive at time t is therefore equal to $n \times m$.

Each individual agent alive at time t, $A_{ij}(t)$, is completely characterized by a *forecast rule* for future prices. For implementation of the genetic operators of the genetic algorithm it is necessary to code each of these forecast rules as strings of L binary digits (bits), that is, values chosen from the binary alphabet $\{0, 1\}$. A single string—agent $A_{ij}(t)$'s forecast rule—might look like this:

$$A_{ij}(t) = 1001101010.$$

[9] There is another, more practical, difficulty with a non–overlapping information structure. Consider the n-period model, where n is a large number. Virtually all of the agents alive at date t were also alive at date $t-1$. If the only way to inject new forecast rules into the system is through a single, "newborn" generation of agents, then the fraction of new, updated decision rules at each date t will only be $1/n$. The injection of new decision rules through the creation of offspring thus appears to be a rather poor device, especially if the goal is to design a system that adapts quickly to new circumstances. The result of the emulation procedure is that in every period outside of equilibrium, almost all agents have an incentive to alter their forecast rules, so that the fraction of new, updated forecast rules may be as high as $(n-1)/n$.

The binary representations of agents' forecast rules can be decoded as follows. The first $L-1$ bits reading from left to right specify whether or not to include certain lagged values in either a linear least–squares autoregression on past price levels or a linear least–squares autoregression on past first differences in prices. If the bit in position ℓ is set to 1, then the ℓ^{th} lagged value is to be included in the autoregression; if it is set to 0, then the ℓ^{th} lagged value is not included in the autoregression. The last bit (the bit furthest to the right) specifies whether the autoregression is to be performed using price levels or first differences in prices. If this last bit is set to 1, then the forecast rule involves an autoregression of price differences on lagged price differences. If it is set to 0, the rule specifies an autoregression of price levels on past price levels. A constant term is always included in the specification of any autoregression.[10]

Once the forecast rule specification is decoded, the model parameters are estimated using the first half of the available data on past prices.[11] The fitted forecast model is used to generate a sequence of point forecasts for price levels over the period of the second half of the available data on past prices.[12] The criterion used to assess forecast accuracy is then the mean squared error (MSE) between the model forecasts and the actual data over the second half of the price data set. Each forecast rule's *fitness value* is simply the inverse of its MSE based on this calculation. In the genetic algorithm, rules with higher fitness will be more likely to be chosen by agents for the purpose of forecasting future inflation.

Almost every member of this class of forecast rules will accurately forecast inflation at either of the two stationary perfect foresight equilibria.[13] Outside of equilibrium, however, different forecast rules in the class will often yield very different forecasts of future prices based on the same data.[14]

[10] As an example, consider $A_{ij}(t) = 1001101010$. The rightmost bit, which is set to zero, indicates that the forecast rule will involve an autoregression of the price level $P(t)$ on lagged values of the price level, $P(t-j)$. The first $L-1=9$ bits, starting from the left, provide the specification of this agent's linear autoregressive forecast rule which we can write as

$$P(t) = \gamma_0 + \gamma_1 P(t-1) + \gamma_2 P(t-4) + \gamma_3 P(t-5) + \gamma_4 P(t-7) + \gamma_5 P(t-9),$$

where the γ terms denote the coefficients to be estimated by fitting this model to the data set on past prices. If the rightmost bit had been set to one rather than zero, then the forecast rule specification would instead have been:

$$\Delta P(t) = \gamma_0 + \gamma_1 \Delta P(t-1) + \gamma_2 \Delta P(t-4) + \gamma_3 \Delta P(t-5) + \gamma_4 \Delta P(t-7) + \gamma_5 \Delta P(t-9),$$

where $\Delta P(t-j) = P(t-j) - P(t-j-1)\ \forall j$.

[11] We always insure that there is enough data to run the most demanding regression in the first period.

[12] We did not allow the agents to forecast negative prices, even if that is what their forecast rule prescribed. We substituted a lower bound, a very small positive number, for a negative price forecast.

[13] The only forecast rule specification that is feasible but clearly inaccurate in an equilibrium with positive inflation is a string where every bit is set equal to 0. Such a rule would involve a regression of the price level $P(t)$ on a constant alone, and could not capture the inflationary trend.

[14] For one fairly trivial example, if the price level was relatively high in some period $t-j$, compared with other recent periods, a time t forecast rule that includes this jth lagged value of the price level or of the difference in prices can lead to substantially different forecasts of future rates of return relative to other rules that do not include this

This represents our claim that when we initialize our systems randomly, as we will in the implementation, we are considering systems with a initially rich diversity of beliefs concerning future prices. These agents with diverse opinions therefore take quite different actions with respect to holding fiat currency. In addition, we allow for a sufficiently large set of rules to make the problem of choosing from among the various rules interesting and nontrivial. In our implementation, we chose a string length of $L = 21$, so that there were in fact 2^{21}, or approximately 2.1 million different forecast rules that agents could choose from. This large set of rules helps to justify our use of a genetic–based search algorithm; the genetic algorithm is very adept at searching large solution spaces because of its use of a population of candidate strings to process many different solutions simultaneously, in parallel.

3.3 How forecasts are used

An agent is completely described by a forecast rule. At each date t, agents fit their rules to the set of past price data using the entire history of prices through period $t-1$. Forecasts of future price levels are then generated and converted into forecasts of rates of return—gross inflation factors—over the remainder of the agent's planning horizon. An agent in the kth period of life, where $1 \leq k < n$ will require $n - k$ forecasts of future gross inflation factors in order to make consumption and savings decisions over the remaining periods of life. Given the sequence of inflation forecasts generated by each forecast rule, each agent solves the optimization problem stated in section 2, defined over the remaining number of periods in their lives and using the given sequence of endowments that is relevant to their planning horizon.[15] The solution to the optimization problem results in a sequence of consumption and savings amounts that is optimal for each agent given the forecasts generated by the rule they are using a time t.

We are particularly interested in each agent's optimal savings decision in period t, which we denote by $s_{ij}(t)$. By summing together all of these savings decisions, we are able to calculate aggregate savings in the economy at time t:

$$S(t) = \sum_{i=1}^{n-1} \sum_{j=1}^{m} s_{ij}(t).$$

jth lagged value.

[15] We note that while agents in our model may lack perfect foresight knowledge of future prices, they do know how to solve a constrained optimization problem. Thus, we depart from previous analyses involving genetic algorithm learning (e.g. Arifovic (1995, 1996)) in that the agents in our model are not learning how to optimize, but are rather learning about which forecast rules provide good forecasts.

Using this value together with aggregate savings from the previous period, $S(t-1)$, and the previous price level, $P(t-1)$, we can generate the realization for the price level at time t, $P(t)$, according to the law of motion (3). Each new price level realization gets added to the data set on past prices. We now turn to a description of the process by which the set of forecast rules is updated following each new realization of the price level.

3.4 Genetic operators

Prior to making consumption and savings decisions in the first period of life, the members of the "newborn" generation have to adopt forecast rules that they can use to forecast future gross inflation. We imagine that these newborn agents construct forecast rules based on an exchange of ideas with members of the current population of agents. We do this by using a series of genetic operators: reproduction, recombination, mutation and election. These operators are all part of the genetic algorithm.

The process of attaching a forecast rule to a newborn agent begins with the first genetic operator, *reproduction*. The reproduction operator involves choosing two agents (forecast rules) randomly, with replacement, from the entire $n \times m$ population of agents alive at time $t-1$.[16] One can think of each newborn agent as meeting with two other agents who have been alive for one or more periods, and considering each of these two agents' forecast rules for adoption. The newborn calculates the fitness of the two forecast rules and ranks the one with highest fitness as the candidate for adoption. If reproduction were the only operator, the newborn would adopt this forecast rule with the highest fitness value, and, repeating the process for all newborn agents, the forecast rules with higher fitness would tend to be adopted by the newborn generation. Reproduction provides most of the evolutionary pressure in our model. But other operators are necessary in order to introduce new forecast rules into the population.

To do this, we imagine that the newborn agent uses the two ranked forecast rules from the reproduction operator to develop two *alternative* forecast rules. The creation of these two alternative forecast rules involves the use of the genetic operators of crossover (recombination) and mutation. The crossover operator involves taking the two strings and swapping portions of the elements of these two strings. To illustrate, suppose that our newborn agent meets with two agents

[16] The fact that we create newborns from the entire stock of genetic information available at time t, instead of just from the oldest generation (as in Arifovic (1995)), is what we call *mixing*. Mixing substantially reduces the potential of the system to cycle near the steady state.

(already alive in the population) who have the following forecast rules: 010100110010111010101 and 111001010101101101010. With fixed probability p^c, an integer is chosen from the interval $[1, L-1]$. Let us suppose that an integer is chosen, and it turns out to be 12. In this case, the two strings are each divided into two parts with the dividing point to the right of bit position 12:

010100110010 | 111010101

111001010101 | 101101010

The two parts to the right of bit position 12 are then swapped. The two resulting strings,

010100110010101101010

111001010101111010101

may take the place of the two strings in the newborn population; for the moment we simply set them aside. With probability $1-p^c$, no integer is chosen, and crossover is not performed on the randomly paired agents' forecast rules; in this case, the two post–crossover strings are the same as the two pre–crossover strings. The purpose of crossover is to combine blocks of bits with other blocks of bits—Holland (1975) calls these blocks *schema*—in an effort to advance the search for improved forecast rules. The rationale is that schema (blocks of bits) that are highly fit will not only survive into successive generations but will also spread throughout the population. The other genetic operator—*mutation*—also serves to advance the search for better forecast rules. The mutation operator is applied to each bit in each of the two recombined, post-crossover strings. With fixed probability p^m, the bit is changed: if the bit was a 0, it is changed to a 1, and if the bit was a 1 it is changed to a 0. With probability $1-p^m$, the bit is left unchanged. The newborn agent then calculates the fitness of the two alternative strings and ranks the string with the highest fitness as a candidate for adoption.

The newborn agent now has two candidate strings for adoption: the more highly fit of the two strings produced by the reproduction operator, and the more highly fit of the two alternative strings produced by the crossover and mutation operators. We use the *election* operator to allow the newborn to make the decision about which one to adopt.[17] Election simply means that the agent will adopt the candidate string with the highest fitness. The idea of election is to allow agents to consider alternative forecast rules, possibly ones that have never been in the population before,

[17]The election operator was first proposed and used by Arifovic (1995).

without forcing them to adopt "bad ideas," forecast rules with fitness worse that they could have had by simply copying an existing rule.

Altogether, then, we imagine that each newborn agent meets with two older agents, experiments with their rules by combining and mutating them, and then chooses the best forecast rule of the four based on past forecast accuracy.[18] We interpret this process as one by which young agents learn from their elders and also inject new ideas into the society. The basic idea of the genetic algorithm is to advance the search for highly fit strings. The reproduction operator serves to copy existing forecast rules so that the building blocks for new forecast rules are rules that have already been used by the population. The crossover and mutation operators allow agents to experiment with new forecast rules, some of which may be very different from those used by members of the current population. While agents experiment with new possibilities for their forecast rules, they also avoid adoption of "bad" rules—forecast rules with high MSEs—an effect which is due to the election operator.[19]

Once the m members of the newborn generation have chosen their forecast rules, these newborn agents emerge at time t as members of the youngest generation—generation 1. It is at this point that the oldest generation—generation n—ceases to exist. The m members of generations $i = 1, 2, ..., n-1$ alive at time $t-1$ are redesignated as members of generations $i = 2, 3, ..., n$ at time t. As we have emphasized, however, it is important that the $m(n-1)$ members of these "middle" generations also update their forecast rules over the remaining horizons of their lives, since in every period there is a new realization of the price level.

3.5 Emulation

We now want to think of the agents who are members of these middle generations as also searching for new ideas about how to improve their forecast rules in the next period of their lives. The search by members of the middle generations for new and improved forecast rules constitutes the *emulation* part of our model. These "middle–aged" agents already have a forecast rule—they are not "blank slates" like the newborns. Thus we imagine that these middle–aged agents continue to consider their current forecast rule as a candidate for the rule they will choose to use in the next period of life.

[18]The 'tournament' selection process we employ has several advantages over the "biased roulette wheel" selection method used by Arifovic (1995) and advocated by Goldberg (1989). See Fogel (1994) for a discussion of the problems with roulette wheel selection.

[19]The fact that agents consider alternatives but do not necessarily adopt them is an important feature of our model because it means that agents are still experimenting with alternative forecast rules even after the system has converged to a steady state. Thus the system has the potential to react effectively to some change in the environment, such as a perturbation of a model parameter.

However, they also explore new possibilities.

In particular, each of the m members of every generation i, $1 \leq i < n-1$ meets one randomly chosen member from the entire $n \times m$ population of agents. The random selection of another agent from the population is made with replacement. Each middle–aged agent considers adopting—emulating—the forecast rule of this randomly chosen 'partner agent.' The middle-aged agents calculate the fitness of their existing forecast rules as well as the fitness of their partner agents' forecast rules. For each agent, the forecast rule with higher fitness is a candidate for use in the next period of life. The middle–aged agents also experiment with new forecast rules. They take their own forecast rule and the forecast rule of their randomly selected partner agent, and they apply the genetic operators of crossover and mutation in order to create two alternative forecast rules in the same manner as was described for the newborn agents. They then calculate the fitness of the two alternatives, and the better of the two becomes a candidate for adoption in the next period of life. We then employ the election operator to allow the agents to choose to adopt the better of the two candidate forecast rules. The result is that each middle–aged agent has one, possibly new, forecast rule for purposes of decision–making in the next period of life.[20]

We interpret this process as one where middle–aged agents exchange ideas with other members of society. At the same time, they experiment with new possibilities through crossover and mutation, in order to try to improve the accuracy of their forecast rules and therefore increase their utility by making good consumption and savings decisions. While middle–aged agents may choose to emulate the forecast rule of their partner agent, or may have chosen to adopt one of the two alternative rules they created via experimentation (crossover and mutation), a third possibility is that these middle–aged agents continued to use the same forecast rule they used in the previous period.

We note that forecast rules created via this emulation process may never have appeared before in the population of agents. These new decision rules are in addition to those created by the newborn agents, and provide another source of "new ideas" in every period. The number of new forecast rules per period due to emulation may be as high as $m(n-2)$, while the number of new forecast rules per period that can be attributed to the newborn generation can be no higher than m. Thus, the emulation feature of our model can inject many more new ideas into the system relative to the number of new ideas provided by the newborn generation alone.

[20] Agents who are members of generation $n-1$ in period $t-1$ do not engage in emulation. Their planning horizon consists of only one period, the last period of their lives, and consumption in the last period of life is predetermined as 100 percent of wealth, so there is no need to experiment with forecast rules.

3.6 Summary of the algorithm

Before turning to the results, we provide a brief summary of our learning and emulation algorithm. Given a population of strings at time $t-1$, our learning and emulation algorithm operates on this population of agents in the following sequence of steps:

1. A set of m newborns enters the model without forecast rules. These agents decide on a forecast rule by "exchanging ideas" with some of the members of the population currently alive. This process is implemented via the application of the genetic operators reproduction, crossover, mutation, and election. The result is m newborn agents.

2. Emulation takes place. Agents of all generations A_{ij}, $i < n-1$ are randomly paired with one of the members of the population. The middle-aged agents consider copying their partner's forecast rule and also consider new ideas through application of the genetic operators crossover and mutation. The result is an updating of the $m \times (n-2)$ forecast rules associated with middle-aged agents.

3. Time officially changes from date $t-1$ to date t. The m members of the newborn population become the m members of generation 1. Agents A_{ij}, $i < n$ enter the next stage of life. Agents of the oldest generation, A_{nj}, cease to exist.

4. Aggregate savings is determined. Each agent A_{ij}, $i < n$ fits his forecast rule using the entire history of prices through time $t-1$. The fitted forecast rule is then used to generate a sequence of forecasts of the gross inflation factor over the relevant planning horizon of agent A_{ij}. Given these forecast values, the agent solves the optimization problem over the remaining period of his life. This results in a savings decision for period t, denoted by $s_{ij}(t)$. Aggregate savings is then given by $S(t) = \sum_{i=1}^{n-1} \sum_{j=1}^{m} s_{ij}(t)$.

5. Using aggregate savings at dates $t-1$ and t, a new market price level realization, $P(t)$, is determined according to equation (3).

6. This process is repeated to some maximum number of iterations.

This completes our description of the sequence of steps involved in our model of learning and emulation. We now turn to the design and results of our computational experiments.

4 Computational experiments

4.1 Parameterization and initialization

The endowment sequence we chose is based on a 55–period (annual) model where the endowment in period k is given by $w_k = 56 - k$. Conversion from a 55–period model to an $n < 55$ period model was accomplished by integrating this endowment sequence using appropriate limits on the integration. We then normalized the condensed profile so that the first period endowment was always equal to unity.[21] By following this procedure, we ensure that the endowment profile is consistent across the sequence of n–period economies that we study.

We chose values for the parameters of our learning and emulation model based on results in the artificial intelligence literature. We maintained a constant total population size, ps, of approximately 60 agents where possible. The number of agents in each generation was set in most cases according to the formula $m = ps/n$. When we reached $n = 7$, the highest value of n that we considered, we chose to set $ps = 56$, so that $m = 8$.

We set the bitstring length $L = 21$, creating a set of about 2.1 million possible forecast rules. The probability of mutation p^m was set equal to the inverse of the string length, $1/21 \approx .048$. A low probability of mutation, in particular, a probability that is inversely proportional to the string length, is standard in the artificial intelligence literature. There is less consensus regarding the choice for the probability of crossover, p^c, although it is generally believed that a genetic algorithm is most effective with $p^c \geq 0.5$. For our simulations, we chose to set p^c equal to 1.0.[22]

The initial population of agents (forecast rules) in each simulation was randomly generated; each bit in every bitstring of the initial $n \times m$ population of bitstrings was assigned the value 0 or 1 with probability .5. We also began each simulation with an initial set of past market prices which were chosen as follows. First, given the model parameters (discussed below), we numerically solved for the two stationary inflation factors, β_ℓ and β_h. We then chose random values for past gross inflation over the uniform interval $[\beta_\ell, \beta_h]$; this procedure ensures that past inflation has been in a range that is theoretically plausible given the particular parameterization of the model. We then converted these randomly chosen inflation factors into a sequence of past market prices. This initial

[21]For example, when $n = 3$, we integrate the formula $w_k = 56 - k$ from 1 to 19.33, then from 19.33 to 37.67, and finally from 37.67 to 56. Normalizing the first period endowment to unity, this process yields a three period endowment sequence of $\{1.0, 0.6, 0.2\}$.

[22]These parameter values are consistent with those suggested by Grefenstette (1986) and Goldberg (1989). We note that since the election operator assures that bad forecast rules will not be adopted, agents can experiment with alternatives through crossover without penalty.

set of market prices was made large enough to accommodate initial forecast rules involving up to 20 lagged values, and also allows for the testing of these rules so that MSEs on the initial population of forecast rules can be calculated.

The goal of our computational experiments was to assess whether the population of artificially adaptive agents could learn to create and adopt forecast rules that are consistent with perfect foresight knowledge of future prices. Since there are always just two stationary perfect foresight equilibria in this model, we were also interested in which of the two (if any) the population of agents might settle on. Since the population of initial forecast rules and the initial set of prices were randomly generated, our experiments can be viewed as a test of whether either perfect foresight steady state is *globally* stable under an evolutionary learning scheme.

4.2 Satisfaction of the government's budget constraint

Outside of a stationary equilibrium, where agents have not yet discovered and coordinated upon a forecast rule that is consistent with perfect foresight knowledge of future prices, it can be the case that some or all of the agents' forecast rules yield forecasts of inflation that are so high that these agents optimally choose not to hold any fiat currency; that is, they optimally choose not to save, and instead they simply consume their endowments. If enough of the forecast rules in the population lead to such zero (or near zero) savings decisions, the government's deficit borrowing constraint, as given in equation (1), may fail to hold. We interpret this situation as one of *currency collapse*.

While we allow for the possibility of such currency collapse outcomes, we also wanted to observe what happened to the evolution of the system after such collapses had occurred. We therefore implemented the following taxation scheme. After aggregate savings has been determined, but prior to the end of the period, the government checks that the level of aggregate savings is sufficient to meet its revenue requirements according to equation (1). If aggregate savings is too low, the government computes the extent of the shortfall and forces each agent (regardless of their savings decision) to give up some small amount of the consumption good and instead hold some small amount of currency. The government chooses the amount of the tax that all agents must pay so that summing up across all agents, the government is able to make up its shortfall plus a small additional amount which ensures that the deficit financing constraint is more than exactly satisfied. The imposition of the tax prevents a currency collapse outcome from violating the government's budget constraint, which would lead to a breakdown of the law of motion for prices (equation (3)) and cause our algorithm to terminate following the first period of the currency collapse outcome. The taxation scheme thus creates a third stationary outcome for the model, which is associated with near–violation of the government budget constraint. We chose the definition of "near–violation" to be so close to violating the budget constraint that the associated stationary inflation rate was one to two orders of magnitude greater than β_h.

Our taxation scheme is a simple device that allows our system to continue evolving regardless of the forecasts being made by the agents. In the description of our results (section 4) we report the number of times, if any, that the tax was imposed. One can then interpret any computational experiment in which the tax was imposed as a currency collapse outcome. However, since our taxation scheme allows the system to continue following a currency collapse we can continue to characterize the evolution of our system beyond any currency collapse outcome. Following a currency collapse we can observe one of two further developments. The first is that a number of agents choose to adopt forecast rules that forecast lower inflation and using these rules they choose to save enough so that the level of aggregate savings exceeds the level necessary to meet the government's revenue requirements. In this case, the system can continue indefinitely without further imposition of the tax, possibly converging eventually to one of the stationary perfect foresight equilibria of the model. We observed this type of outcome in our computational experiments. On the other hand, some or all agents may continue to forecast very high inflation so that the taxation scheme continues to be implemented every period. In this case, the system can remain in a state of persistent currency collapse. We observed this outcome as well.

4.3 Design of computational experiments

Computational experiments with our learning and emulation algorithm were conducted in a sequence of n–period endowment overlapping generations economies. The model parameters that we allowed to vary were the number of periods in agents' lives, n, and the level of the government's deficit, ξ. In particular, the number of periods in agents' lives was allowed to vary from $n = 3$ on up to $n = 7$. The level of the government's deficit was chosen as a certain fraction, $x \in (0, 1)$, of the maximum feasible deficit level $\bar{\xi}$, that is, $\xi = x\bar{\xi}$. For each of the 5 different values for n, we considered cases where $x = .1$, $.3$ and $.5$. We used our model of learning and emulation to perform 10 simulations for each of these 15 model economies for a total of 150 computational experiments. Each simulation ran for 125 iterations.[23]

4.4 Simulation results

4.4.1 General summary

Our computational experiments yielded three qualitatively different types of outcomes. The most common outcome was that the population of agents eventually achieved coordination on the low inflation, stationary perfect foresight equilibrium of the model. That is, all agents' forecast rules eventually yielded forecasts for gross inflation that were equal to the value of β_ℓ for the given model economy. In such cases, the forecasts of inflation became self–fulfilling in that agents could not do better (in terms of lower MSE) by shifting to other forecast rules. In tandem with the low inflation outcome, we never observed coordination on the stationary perfect foresight equilibrium with relatively high inflation, β_h. This finding is consistent with earlier analyses of learning behavior in the context of two period overlapping generations economies.

The second qualitative outcome that we observed was that of a persistent, self–fulfilling currency collapse. In this case, at the end of the computational experiment all (or almost all) agents' forecast rules were forecasting very high inflation causing agents to save little or nothing. Given these savings decisions, actual inflation tended to be high, and so the expectations of high inflation became self–fulfilling. The government's taxation scheme was imposed in every period, and the system never deviated away from experiencing the very high rates of inflation associated with the taxation scheme. We refer to this outcome as a persistent currency collapse because in the absence of the government's taxation scheme fiat currency would have ceased to have any value.

The intuition behind the persistent currency collapse outcome is fairly straightforward. It can happen that, by chance, some agents' forecast rules yield forecasts of very rapid increases in the price level, and hence very high inflation rates. These agents then decide to hold little or no currency, and in particular, less currency than they held in the previous period. If we hold the other agents' decisions constant, equation (3) indicates that this effect tends to push the next period's inflation rate higher. The genetic algorithm then tends to reward (assign higher fitness to) the forecast rules that forecast higher inflation. This means that the forecast rules belonging to these agents tend to be copied and propagated in the economy. This process can lead to more agents forecasting high

[23] Data accumulates in the algorithm, and since a great number of regressions have to be run during each iteration, we found significant slowing of the algorithm as the number of iterations increased. This effect is somewhat offset by the tendency of the algorithm to create similar populations, in which case certain evaluations do not have to be done multiple times. Generally, however, our experimental design exhausted our available computational resources.

inflation, these agents then choosing to hold little or no currency, actual inflation rising again, and still more agents choosing rules that forecast high inflation. In this way, a self-fulfilling hyperinflation can be achieved and sustained.

A third outcome was that the population of agents failed to coordinate on either of the two stationary perfect foresight equilibria within the allotted number of iterations. This failure–to–converge outcome does not imply that the system would not have eventually converged to a stationary equilibrium if the system had been allowed to run for a longer number of periods. We simply note that after 125 iterations, the system had not achieved anything resembling convergence.

4.4.2 Detailed summary

A detailed summary of the main results from our 150 computational experiments is provided in Table 1, which is partitioned according to the value of n. For each different value for n, $n = 3,4,5,6,7$, there were three different values for the fraction of the maximum deficit, $x = .1,.3,.5$. The left column of Table 1 reports the various different combinations of n and x that we considered in our simulation analysis and the associated values for β_ℓ and β_h. To the right of this column are various summary statistics on each of the replications (numbered 1 through 10) that we conducted for each (n, x) pair.

The first statistics that are reported are the mean, μ_{10}, and standard deviation, σ_{10}, of the actual gross inflation factor, β, from the last 10 iterations of each experiment. These statistics can be used to assess whether the system has achieved either of the two stationary perfect foresight equilibrium values, β_ℓ or β_h. For example, in the case where $n = 3$ and $x = .1$, we see that in experiment 1, $\mu_{10} = 1.019$ and $\sigma_{10} = 0.000$. For this (n, x) pair, the value for $\beta_\ell = 1.019$. We may therefore conclude that the system did coordinate on the low inflation, stationary perfect foresight equilibrium of the model within 125 iterations. The σ_{125} statistic is the standard deviation of β over all 125 iterations of each experiment. This statistic provides a characterization of the volatility of the actual gross inflation factor over the 125 periods of each simulation. Generally, we found gross inflation factors could be quite volatile in the course of our experiments, with the level of inflation often many orders of magnitude higher than levels consistent with stationary equilibria, even in cases where coordination on a stationary perfect foresight outcome eventually attained.

The numbers under the heading $\bar{H}$ represent the average of all of the *Hamming distances* calculated for all pairs of strings in the population at the end of each experiment. The Hamming distance between any two binary strings provides a measure of the extent to which the two strings differ from one another; this metric can be defined as follows. Let $\vec{a}$ denote a binary string of length L with elements $(a_1, a_2, ...a_L)$, $a_i \in \{0,1\}$ and let $\vec{b}$ denote another binary string of length L with elements $(b_1, b_2, ...b_L)$, $b_i \in \{0,1\}$. Then the Hamming distance between these two strings is defined by:

$$H_{\vec{a},\vec{b}} = \sum_{i=1}^{L} |a_i - b_i|.$$

Note that if two strings are identical, their Hamming distance is zero, while the maximum Hamming distance between two L–length strings is L. In our simulations we had population sizes of $N = 60$ or $N = 56$ strings. We therefore calculated $N(N-1)/2$ Hamming distances for all possible pairings of strings that were in the population following iteration 125 of each simulation. The *average value* of all of these Hamming distance calculations is the number reported under the heading $\bar{H}$. If $\bar{H} = 0$, then all N strings were identical at the end of the simulation. The maximum value for $\bar{H}$ is given by the formula $\frac{NL}{2(N-1)}$. Since we used a string length $L = 21$, this maximum value for $\bar{H}$ is 10.68 when $N = 60$ and 10.69 when $N = 56$.

Finally, the entries under the heading cc indicate the number of periods in each experiment (out of 125) in which a *currency collapse* occurred—that is, the number of times the tax scheme that ensures satisfaction of the government's budget constraint had to be imposed. One can simply interpret any experiment in which this entry is not zero as a currency collapse outcome. If we take the tax device more literally, we note that one or more periods of currency collapse need not prevent the system from ultimately achieving coordination on the low inflation stationary perfect foresight equilibrium. Notice also that there are several instances of persistent currency collapse, where agents are continually forecasting high inflation and the government is forced to impose its taxation scheme in every period. These instances can be identified in Table 1 as experiments where μ_{10} is greater than β_ℓ and β_h by one or two orders of magnitude, and in addition $\sigma_{10} = 0.0000$ and $cc > 10$. As an example, see experiment #1 for the case where $n = 5$ and $x = .5$. The value for μ_{10} in these cases is the stationary inflation factor associated with the case where all agents optimally choose to save zero, and all government revenue is derived from the imposition of the taxation scheme.

The statistics in Table 1 provide us with some sense of relative likelihood of the three observed outcomes as we vary both n and x. We note that for relatively low values of n (and any value for x), the system is more likely to achieve coordination on the low inflation stationary perfect foresight equilibrium where $\beta = \beta_\ell$. Indeed as n increases we see that persistent currency collapse outcomes become increasingly likely. One can interpret this as less tendency for the system to coordinate on the low inflation steady state when agents are allowed to change their forecast rules more and more often during their lifetimes. We also observe that for a given value of n, convergence to the low inflation stationary perfect foresight equilibrium appears to be more likely for relatively lower values of x than for relatively higher values of x. Indeed, the average number of periods of currency collapse is highest when $x = .5$, and most of the self–fulfilling currency collapse situations occur when $x = .5$. This can be interpreted as saying that as the government presses toward the maximum amount of revenue from seignorage, the risk of destabilizing the economy increases. Interestingly, the government does not have to attempt an unsustainable policy ($\xi > \bar{\xi}$, a case where no rational expectations equilibria exist) to run this risk. Merely getting relatively close to such a policy, but keeping $\xi < \bar{\xi}$, is sufficient.

4.4.3 Some illustrations

Figures 1, 2 and 3 provide illustrations of the three different qualitative outcomes that we observed. Each figure plots the evolution of the gross inflation factor (β) over the 125 iterations of three different experiments. To make the comparison as simple as possible we considered three experiments from the same (n, x) pair—all three experiments were for the case where $n = 5$ and $x = .5$. We stress that the volatility of gross inflation tended to be surprisingly high in these experiments, as evidenced by our use of logarithmic scaling in the figures.

Figure 1 is an illustration of one experiment (#7) where the system achieved coordination on the stationary perfect foresight equilibrium, $\beta_\ell = 1.069$, within the allotted 125 iterations. As Table 1 indicates, the mean and standard deviation of the last 10 values for β in this experiment were $\mu_{10} = 1.069$ and $\sigma_{10} = 0.000$. Furthermore, there were no instances of any currency collapse along the way. The standard deviation in β over all 125 iterations was $\sigma_{125} = 0.591$; as Figure 1 shows, the system never deviated very far from the low inflation stationary equilibrium. However, there are other examples where the system ultimately converged upon the low inflation stationary equilibrium, but where some episodes of currency collapse were experienced along the way.

Figure 2 illustrates the path of β for another experiment (#1) involving the same model economy. Here we see that the path of the gross inflation factor β starts to become rather volatile after only a few iterations. The first currency collapse occurs in period 56 when inflation soars to nearly 4,000 percent.[24] Aggregate savings falls to a very low level, and the government is forced to require

[24]Currency collapse episodes may be identified in these figures by a discontinuity in the line connecting gross inflation factors from one period to the next.

agents to hold some fiat currency to meet its budget constraint. Following the currency collapse and the government's implementation of the tax scheme, gross inflation returns to relatively low levels only to eventually rise once again to very high levels precipitating another currency collapse. This same pattern is repeated several more times before the currency collapse outcome becomes a self–fulfilling event beginning with period 84. Thereafter, the gross inflation factor remains approximately constant at the level 125.3. This is the gross inflation factor that is compatible with the government meeting its budget constraint in every period by requiring all agents to hold some small amount of fiat currency.

Finally, Figure 3 illustrates another experiment (#8) involving the same model economy where the value of β had failed to achieve any kind of convergence after 125 iterations. The evolution of this system is characterized by numerous currency collapse episodes; Table 1 documents that currency collapses are experienced in 16 of 125 periods. Nevertheless, no sustained currency collapse is observed within the 125 iterations. While it is not clear what would happen to this system if it were allowed to run for a longer length of time, we speculate that it would either eventually converge to the low inflation stationary equilibrium or it would end up at a self–fulfilling currency collapse outcome.

4.4.4 Evolution of forecast rules

In addition to considering the behavior of the gross inflation factor over 125 iterations, it is also of interest to consider the evolution of the population of forecast rules. The initial population of rules was randomly generated. The average Hamming distance between all pairings of the initial randomly generated forecast rules should be approximately one–half the stringlength: $\bar{H} = L/2 = 21/2$, or 10.5. In most cases, the final populations of rules, following iteration 125, are considerably less heterogeneous, judging from the average Hamming distances ($\bar{H}$) reported in Table 1.

Looking at these average Hamming distances we see that with a single exception, $\bar{H}$ is never equal to zero, indicating that after 125 iterations, there always remains some heterogeneity in the specification of agents' forecast rules, even in cases where the system has converged to a stationary outcome. This is not a surprising finding, however, since nearly all the forecast rules that agents can use will accurately forecast future prices when the system has achieved a stationary perfect foresight equilibrium. Thus, there is no reason that forecast rules should all be equivalent; if all rules are accurately forecasting future prices they will have the same MSE values, and will be able to continue to survive as members of the population of rules.[25] In the case of a stationary currency collapse outcome, all forecast rule specifications are capable of generating forecasts of very high inflation, so the agent's choice of which rule to use in such cases is not a very important consideration. For these reasons, there does not appear to be, nor do we expect, any consistent relation between the model parameters that we vary (n or x) and the value of $\bar{H}$.[26]

5 Conclusion

We have interpreted our model of learning and emulation as describing a process by which heterogeneous agents in a large competitive economy exchange and experiment with new ideas about how to forecast the future. Newborn agents form forecast rules by communicating with some members of the population they are about to enter and by experimenting with new possibilities. Middle–aged agents also exchange ideas with other agents in the population and experiment with alternative forecast rules. These middle–aged agents may adopt an alternative forecast rule if they find that it yields a lower MSE relative to their existing forecast rule. Thus our model envisions agents both emulating others and also contemplating new, untried possibilities. We think that this model provides a good description of trial-and-error learning for use in macroeconomic models. We also regard our emulation procedure involving the middle–aged members of the population as an improvement on a simple genetic algorithm approach in the context of our large competitive economy.

Based on a set of computational experiments in which we have considered a variety of model economies, we find three possible long-run outcomes. The first and most common outcome is that the population of agents eventually achieves coordination on the low inflation, stationary perfect foresight equilibrium of the model. This outcome appears to be more likely for low values of n (agents consider changing their forecast rules relatively less often during their lifetimes) and x (the government raises relatively less revenue from seignorage) than for higher values of these parameters. While the agents in our economies often coordinated on the low inflation stationary perfect foresight equilibrium, we never observed coordination on the high inflation stationary perfect foresight equilibrium of the model. Instead, our second outcome is that the system experiences persistent currency collapse. This persistent currency collapse outcome appears to become more likely as n and ξ are increased.

[25] There is no penalty for unparsimonious forecast rule specifications.

[26] We note that agents' choice of a forecast rule specification based on either price levels or on first differences in prices does not appear to have had any effect on whether or not the system converged (or failed to converge) to a stationary outcome. Most final populations of forecast rules included some mixture of price level and first difference specifications.

In this scenario, some agents' forecast rules by chance yield forecasts of very high inflation rates. These agents then decide to hold little or no currency; and this effect tends to push the next period's inflation higher and thus rewards the agents that forecast high inflation. The forecast rules belonging to these agents then tend to be copied and propagated in the economy. In this way, a self-fulfilling hyperinflation can be achieved and sustained. Our third outcome is one of non-convergence—after 125 iterations the dynamics were still inconclusive.

Our environment has many features we think are interesting from the perspective of studying learning in macroeconomic models. In our formulation of the learning problem, the agents are choosing from among a vast set of potential forecast *rules*, as opposed to learning about the correct parameterization of a single forecast rule. We consider the decisions made by a relatively large population of initially heterogeneous agents, all of whom are interacting with one another in competitive general equilibrium environment. The artificial agents in our model, rather than learning how to optimize, are learning which forecast rules yield the best forecasts. We think that these features constitute a plausible formulation of the learning problem that economic agents face. Finally, our multi-period overlapping generations framework begins to place agents in a realistically complex environment, one where expectations have to be formed many periods into the future before decisions can be made today.

References

Andreoni, J and J.H. Miller, (1995), "Auctions with Artificial Adaptive Agents," *Games and Economic Behavior* 10, 39–64.

Arifovic, J. (1994), "Genetic Algorithm Learning and the Cobweb Model," *Journal of Economic Dynamics and Control* 18, 3–28.

Arifovic, J. (1995), "Genetic Algorithms and Inflationary Economies," *Journal of Monetary Economics* 36, 219–243.

Arifovic, J. (1996), "The Behavior of the Exchange Rate in the Genetic Algorithm and Experimental Economies," *Journal of Political Economy* 104, 510–541.

Arthur, W.B. (1994), "Inductive Reasoning and Bounded Rationality" *American Economic Review* 84, 406–411.

Axelrod, R.M. (1987), "The Evolution of Strategies in the Iterated Prisoner's Dilemma," in: L. Davis, ed., *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann Publishers, Los Altos, CA.

Binmore, K.G. and L. Samuelson (1992), "Evolutionary Stability in Repeated Games Played by Finite Automata," *Journal of Economic Theory* 57, 278–305.

Bullard, J. (1992), "Samuelson's Model of Money with $n-$ Period Lifetimes," Federal Reserve Bank of St. Louis, *Review* May/June, 67–82.

Evans, G.W. and G. Ramey (1995), "Expectation Calculation, Hyperinflation and Currency Collapse," in H. Dixon and N. Rankin (eds.), *The New Macroeconomics: Imperfect Markets and Policy Effectiveness*, Cambridge University Press, Cambridge.

Evans, G.W. and G. Ramey (1996), "Calculation Adaptation and Rational Expectations," working paper, University of Oregon and U.C. San Diego.

Fogel, D.B. (1994), "An Introduction to Simulated Evolutionary Optimization," *IEEE Transactions on Neural Networks* 5, 3–14.

Grefenstette, J.J. (1986), "Optimization of Control Parameters for Genetic Algorithms," *IEEE Transactions on Systems, Man, and Cybernetics* 16, 122-28.

Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning* Addison–Wesley, Menlo–Park, CA.

Goldberg, D.E., and K. Deb (1991), "A Comparative Analysis of Selection Schemes Used in Genetic Algorithms," in G.J.E. Rawlins, ed., *Foundations of Genetic Algorithms.* San Mateo, CA: Morgan-Kaufman.

Ho, T–H. (1996), "Finite Automata Play Repeated Prisoner's Dilemma with Information Processing Costs," *Journal of Economic Dynamics and Control* 20, 173–207.

Holland, J.H. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI.

Holland, J.H. and J.H. Miller (1991), "Artificial Adaptive Agents in Economic Theory," *American Economic Review* 81, 365–370.

Marimon, R., E. McGrattan, and T.J. Sargent (1990), "Money as a Medium of Exchange in an Economy with Artificially Intelligent Agents." *Journal of Economic Dynamics and Control,* 14: 329-373.

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

Michalewicz, Z. (1994), *Genetic Algorithms + Data Structures = Evolution Programs*, 2nd Ed., Springer–Verlag, New York.

Miller, J.H. (1996), "The Coevolution of Automata in the Repeated Prisoner's Dilemma," *Journal of Economic Behavior and Organization* 29, 87–112.

Rudolph, G. (1994), "Convergence Analysis of Canonical Genetic Algorithms," *IEEE Transactions on Neural Networks* 5, 96–101.

Rust, J., J.H. Miller, and R. Palmer (1994), "Characterizing Effective Trading Strategies: Insights From a Computerized Double Auction Model," *Journal of Economic Dynamics and Control* 18, 61–96.

Sargent, T.J. (1993), *Bounded Rationality in Macroeconomics*, Oxford University Press, Oxford.

Table 1. Results from Computational Experiments

Experiment	#	μ_{10} (σ_{10})	σ_{125}	$\bar{H}$	cc	#	μ_{10} (σ_{10})	σ_{125}	$\bar{H}$	cc
	Replication					Replication				
$n = 3$	1	1.019 (0.000)	0.141	1.554	0	2	1.019 (0.000)	0.115	1.897	0
$x = .1$	3	1.019 (0.000)	0.490	2.580	0	4	1.019 (0.000)	0.099	0	0
$\beta_\ell = 1.019$	5	6094 (9309)	8407	2.73	32	6	1.019 (0.000)	0.304	0.964	0
$\beta_h = 2.105$	7	1.019 (0.000)	0.254	0.855	0	8	1.019 (0.000)	0.518	1.372	0
	9	1.019 (0.000)	0.212	2.567	0	10	1.019 (0.000)	0.677	2.138	0
$n = 3$	1	1.062 (0.000)	0.169	5.932	0	2	1.062 (0.000)	38.14	7.206	1
$x = .3$	3	1.062 (0.000)	0.141	3.853	0	4	1.062 (0.000)	0.244	5.106	0
$\beta_\ell = 1.062$	5	1.062 (0.000)	0.139	4.681	0	6	1.060 (0.021)	92.22	3.514	2
$\beta_h = 2.019$	7	1.062 (0.000)	0.145	3.482	0	8	1.752 (1.952)	1034	1.597	6
	9	1.062 (0.000)	0.083	8.598	0	10	1.062 (0.001)	7.789	5.578	0
$n = 3$	1	1.115 (0.000)	0.313	7.221	0	2	1587 (1945)	1295	5.907	20
$x = .5$	3	1158 (1771)	920.8	5.522	16	4	1.089 (0.182)	392.2	5.709	11
$\beta_\ell = 1.115$	5	283.0 (707.3)	335.9	0.389	8	6	1.115 (0.000)	0.187	6.314	0
$\beta_h = 1.922$	7	2.577 (3.830)	372.9	6.642	10	8	1098 (1704)	1127	5.850	19
	9	1.115 (0.000)	0.681	7.387	0	10	1.816 (1.146)	189.7	5.472	12

Table 1. Results from Computational Experiments
(continued)

Experiment	#	μ_{10} (σ_{10})	σ_{125}	$\bar{H}$	cc	#	μ_{10} (σ_{10})	σ_{125}	$\bar{H}$	cc
	Replication					Replication				
$n = 4$	1	1.014 (0.000)	1.957	6.567	0	2	1.015 (0.003)	0.568	0.646	0
$x = .1$	3	1.014 (0.000)	0.314	3.860	0	4	1.015 (0.000)	7.739	2.729	0
$\beta_\ell = 1.014$	5	1.014 (0.000)	0.098	0.496	0	6	1.014 (0.001)	1.316	1.185	0
$\beta_h = 1.764$	7	1.014 (0.000)	8.656	3.200	0	8	1.015 (0.001)	28.79	0.347	1
	9	1.014 (0.000)	0.308	3.994	0	10	1.015 (0.000)	0.224	3.820	0
$n = 4$	1	1312 (3263)	1551	8.378	12	2	1.047 (0.000)	0.150	3.638	0
$x = .3$	3	1.047 (0.000)	0.139	7.239	0	4	1.047 (0.000)	0.083	4.558	0
$\beta_\ell = 1.047$	5	1.047 (0.000)	0.468	4.241	0	6	1.047 (0.000)	0.869	7.236	0
$\beta_h = 1.708$	7	1.047 (0.000)	0.071	1.862	0	8	1.047 (0.000)	1.110	3.215	0
	9	1.047 (0.001)	20.34	7.206	0	10	1.047 (0.000)	0.278	5.054	0
$n = 4$	1	35.83 (57.49)	672.3	6.809	15	2	266.0 (716.9)	1066	5.503	7
$x = .5$	3	1.086 (0.000)	0.135	6.912	0	4	1.194 (0.465)	1398	7.596	11
$\beta_\ell = 1.086$	5	1.087 (0.002)	32.40	7.185	1	6	197.1 (433.2)	953.3	6.920	17
$\beta_h = 1.646$	7	1.086 (.0.001)	0.080	6.987	0	8	1.086 (0.000)	1.279	5.289	0
	9	1.086 (0.000)	0.083	6.687	0	10	7327 (11192)	7851	5.030	25

Table 1. Results from Computational Experiments
(continued)

Experiment	Replication					Replication				
	#	μ_{10} (σ_{10})	σ_{125}	$\bar{H}$	cc	#	μ_{10} (σ_{10})	σ_{125}	$\bar{H}$	cc
$n = 5$	1	1.014 (0.088)	0.389	0.390	0	2	1.012 (0.002)	3825	5.733	1
$x = .1$	3	1.014 (0.017)	2104×10^3	6.687	8	4	1.012 (0.007)	1.343	1.709	0
$\beta_\ell = 1.012$	5	2853×10^3 (5726×10^3)	3623×10^3	6.261	11	6	3063×10^3 (6126×10^3)	3601×10^3	5.548	12
$\beta_h = 1.580$	7	1.012 (0.006)	0.169	7.049	0	8	1.012 (0.006)	2.244	4.198	0
	9	1.006 (0.083)	0.745	5.149	0	10	1.013 (0.001)	0.853	4.754	0
$n = 5$	1	1.038 (0.000)	0.100	4.682	0	2	1.038 (0.000)	2.757	1.487	0
$x = .3$	3	1.039 (0.002)	126.8	1.487	1	4	1.038 (0.003)	0.288	3.363	0
$\beta_\ell = 1.038$	5	1.037 (0.000)	0.996	1.235	0	6	1190×10^2 (2382×10^2)	1685×10^2	1.914	17
$\beta_h = 1.540$	7	1.038 (0.000)	0.497	6.627	0	8	1.038 (0.000)	0.219	1.979	0
	9	8.076 (15.68)	4001	5.091	9	10	1.039 (0.005)	5.937	3.146	0
$n = 5$	1	125.3 (0.000)	1461	2.797	49	2	125.3 (0.000)	83.68	3.903	66
$x = .5$	3	1.157 (0.530)	523.4	5.798	10	4	1.057 (0.163)	197.9	5.228	6
$\beta_\ell = 1.069$	5	32.45 (58.82)	3177	4.359	16	6	125.3 (0.000)	153.4	6.433	33
$\beta_h = 1.494$	7	1.069 (0.000)	0.591	6.767	0	8	26.25 (70.58)	805.7	8.123	16
	9	1.068 (0.002)	0.413	7.002	0	10	76.36 (77.33)	717.4	1.372	12

Table 1. Results from Computational Experiments
(continued)

Experiment	Replication #	μ_{10} (σ_{10})	σ_{125}	$\bar{H}$	cc	Replication #	μ_{10} (σ_{10})	σ_{125}	$\bar{H}$	cc
$n = 6$	1	1.008 (0.015)	0.510	8.891	0	2	1.011 (0.001)	0.516	3.971	1
$x = .1$	3	1.638 (2.386)	4513×10^2	6.776	6	4	3536×10^4 (1061×10^5)	9424×10^4	4.606	16
$\beta_\ell = 1.010$	5	1.016 (0.012)	8719×10^1	3.503	6	6	6756×10^4 (1353×10^5)	7730×10^4	7.454	8
$\beta_h = 1.467$	7	1.010 (0.001)	0.672	4.872	8	8	1.010 (0.001)	0.252	8.523	0
	9	1.009 (0.001)	0.347	3.889	0	10	1.102 (0.434)	6549×10^4	2.472	7
$n = 6$	1	1.036 (0.002)	27.95	1.688	2	2	4.321 (6.229)	85.80	7.834	13
$x = .3$	3	1.033 (0.001)	49.86	4.866	4	4	1.030 (0.003)	21.28	7.305	2
$\beta_\ell = 1.031$	5	1.031 (0.001)	0.493	6.362	0	6	1.031 (0.000)	0.915	1.685	0
$\beta_h = 1.435$	7	90.23 (0.000)	131.2	7.336	103	8	1.032 (0.004)	1.068	5.875	0
	9	1.031 (0.008)	60.19	2.437	1	10	1.032 (0.002)	0.962	4.860	0
$n = 6$	1	1.601 (1.182)	107.2	7.491	11	2	149.7 (0.000)	330.8	5.971	52
$x = .5$	3	149.7 (0.000)	74.05	5.864	70	4	1.205 (0.463)	371.8	6.140	11
$\beta_\ell = 1.057$	5	1.059 (0.004)	1.466	6.592	0	6	1.058 (0.011)	3.729	5.164	0
$\beta_h = 1.400$	7	149.7 (0.000)	226.3	3.641	53	8	1.058 (0.000)	0.636	5.990	0
	9	149.7 (0.000)	29.34	5.003	122	10	1.231 (0.731)	0.520	7.183	0

Table 1. Results from Computational Experiments
(continued)

Experiment	#	μ_{10} (σ_{10})	σ_{125}	$\bar{H}$	cc	#	μ_{10} (σ_{10})	σ_{125}	$\bar{H}$	cc
	Replication					Replication				
$n = 7$	1	1.015 (0.011)	162.3	1.038	1	2	1.008 (0.005)	1.344	7.336	0
$x = .1$	3	1.009 (0.002)	1.916	8.791	0	4	1.016 (0.023)	24.49	6.494	0
$\beta_\ell = 1.008$	5	6797×10^1 (2039×10^2)	2114×10^5	4.291	13	6	1.014 (0.012)	1159×10^2	6.900	1
$\beta_h = 1.390$	7	1.009 (0.002)	0.667	5.034	0	8	1.010 (0.013)	0.852	7.671	0
	9	1.010 (0.009)	5.823	4.862	1	10	1.008 (0.000)	4.005	7.598	0
$n = 7$	1	104.7 (0.000)	820.3	8.231	35	2	104.7 (0.000)	36.13	7.482	109
$x = .3$	3	104.7 (0.000)	101.6	7.571	29	4	104.7 (0.000)	73.54	8.428	51
$\beta_\ell = 1.027$	5	1.025 (0.004)	27.90	7.820	3	6	1.026 (0.004)	1.955	9.177	0
$\beta_h = 1.364$	7	104.7 (0.000)	685.2	6.447	53	8	1.026 (0.002)	1.618	7.881	0
	9	1.647 (1.618)	0.645	7.426	0	10	1.029 (0.006)	77.11	6.010	3
$n = 7$	1	173.9 (0.000)	215.4	6.468	42	2	173.9 (0.000)	89.56	6.805	74
$x = .5$	3	173.9 (0.000)	37.46	5.973	122	4	1.049 (0.005)	25.48	7.927	2
$\beta_\ell = 1.049$	5	173.9 (0.000)	88.01	5.044	92	6	173.9 (0.000)	51.17	6.888	113
$\beta_h = 1.335$	7	1.036 (0.109)	63.46	8.145	5	8	173.9 (0.000)	259.0	5.534	82
	9	173.9 (0.000)	51.86	7.179	112	10	173.9 (0.000)	64.24	5.821	104

Figure 1: Convergence to the Low Inflation Equilibrium

n=5, x=.5, experiment # 7

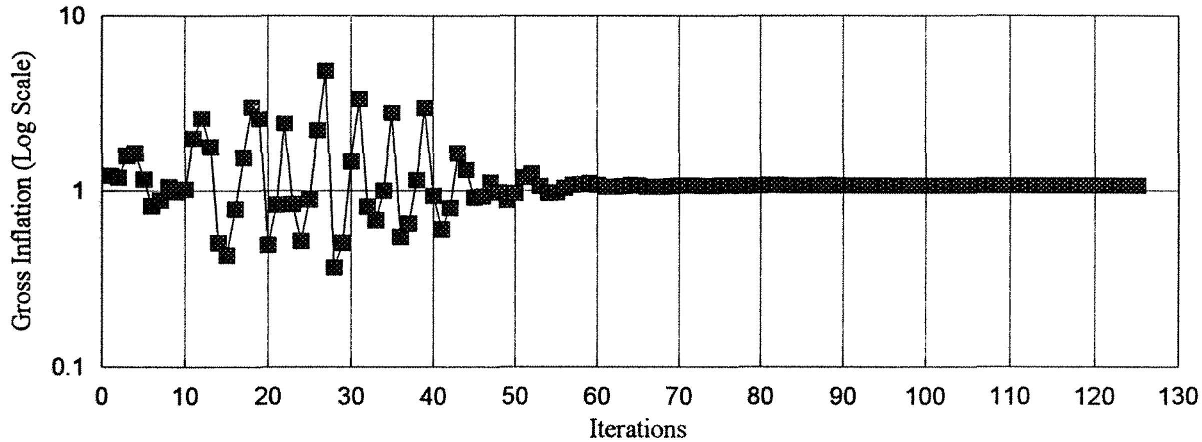

Figure 2: A Persistent Currency Collapse

n=5, x=.5, experiment # 1

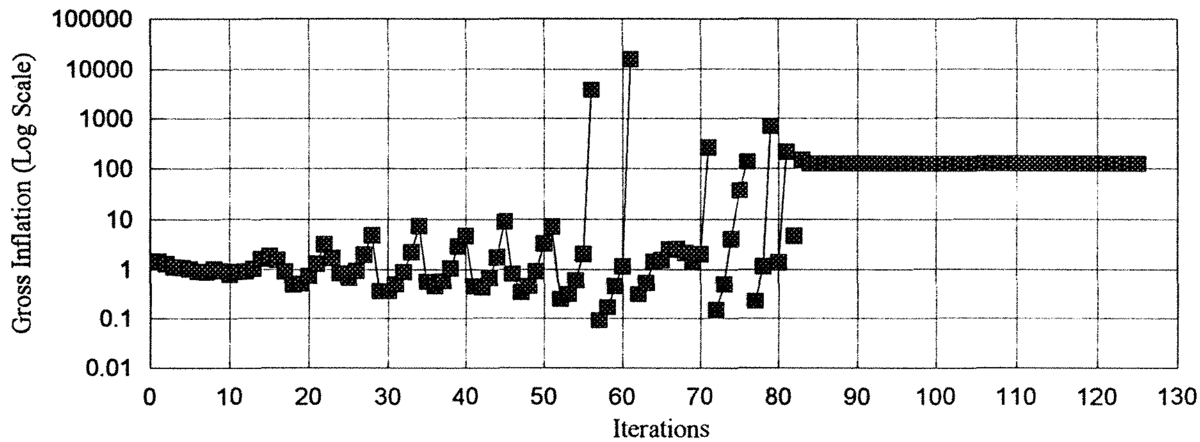

Figure 3: A Nonconvergent Case

n=5, x=.5, experiment #8