Macroeconomic Dynamics, **4**, 2000, 373–414. Printed in the United States of America.

MD SURVEY

EVOLUTIONARY ALGORITHMS IN MACROECONOMIC MODELS

JASMINA ARIFOVIC
Simon Fraser University
and
California Institute of Technology

This paper provides a survey of the applications of evolutionary algorithms in macroeconomic models. Discussion is organized around the issues related to stability of equilibria, equilibrium selection, transitional dynamics, and the long-run evolutionary dynamics different from rational-expectations equilibrium outcomes. The survey also discusses criteria that can be used to evaluate the performance and usefulness of evolutionary algorithms in the macroeconomic context.

Keywords: Learning, Evolutionary Dynamics, Macroeconomic Environments

1. INTRODUCTION

The objective of this survey is to address the issues related to the macroeconomic models in which agents' adaptation is modeled using the evolutionary algorithms: genetic algorithms, classifier systems, and genetic programming. The common feature of these algorithms is the process of adaptation that is based on propagation of decision rules that performed well in the past and on occasional experimentation with new decision rules.[1]

In macroeconomic environments, agents' heterogenous beliefs and decisions affect the levels of endogenously determined prices, which in turn affect agents' payoffs and performance of different decision rules over time. This self-referential character of these economies is the main distinction between these environments and other economic applications of these algorithms.

There are several advantages to modeling adaptation in this way. These algorithms impose low requirement on the computational ability of economic agents. They allow for modeling the heterogeneity of agents' beliefs. Survival of decision rules depends on their performance, measured by the payoff that agents receive by employing them. Also, these algorithms perform better than models with rational agents or alternative models of adaptive behavior in terms of their ability to explain the features observed in experimental economies, as well as some of the features of the actual macroeconomic time series.

I would like to thank Jim Bullard for helpful comments and discussions. Address correspondence to: Jasmina Arifovic, California Institute of Technology, Mail Stop 228-77, Pasadena, CA 91125, USA; e-mail: jasmina@tanis.caltech.edu.

 1365-1005/00 $9.50

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

The research questions addressed by the studies of these algorithms in macroeconomic models can be classified into four categories: (1) the issues related to the convergence and stability of equilibria in the models with unique rational-expectations equilibria, (2) the use of the algorithms as equilibrium selection devices in the models with multiple equlibiria, (3) the examination of transitional dynamics that accompanies the equilibrium selection process, (4) examination of learning dynamics that are intrinsically different from the dynamics of the rational-expectations versions of the models. The exposition in this survey follows this categorization.

The survey also focuses on the methodology for evaluting the usefulness of these algorithms in modeling adaptive behavior. One of the evaluation criteria is the comparison of the predictions of the evolutionary models with those of the rational-expectations models and other learning algorithms. Further, where possible, the evaluation will include the comparison of the features of the time series generated by these algorithms with actual time series. Finally, in case that evidence from the experiments with human subjects exists, the use of experimental data in evaluating evolutionary models also is discussed.

The second section describes applications of the genetic algorithm to the cobweb model, which has a unique rational-expectations equilibrium. The third section gives an overview of applications in models with multiple equilibria, two-period overlapping generations models, n-period overlapping generations model, overlapping generations models with periodic equilibria, a model of growth with increasing returns to scale, and a search model of money. The fourth section gives an overview of the applications that result in the long-run learning dynamics different from the rational-expectations outcomes in an asset-pricing model, a cobweb model, an exchange-rate model, and a model of currency crisis. The fifth section discusses issues related to the interpretation of the results and topics for further research.

2. LEARNABILITY OF EQUILIBRIA

2.1. Cobweb Model

Arifovic (1994) uses the *genetic algorithm* [Holland (1975)] to model adaptation of firms' production decisions in the cobweb model. The model has a unique rational-expectations equilibrium [Muth (1961)]. The learnability and stability of this equilibrium have been investigated using a number of different adaptive algorithms as well as simulating it in the experiments with human subjects. Thus, the results of the genetic algorithm (GA) adaptation can be compared to the behavior of other learning algorithms and evaluated against the experimental evidence.

The GA is a directed stochastic search algorithm based on the mechanics of natural selection. It works with a population of binary strings (chromosomes) that represent possible solutions to the search problem. Each binary string has a fitness value that represents a measure of the performance in a given environment. A population of binary strings is updated using reproduction, crossover, and mutation. Reproduction makes copies of binary strings and does it in such a way that, over time, binary strings with higher fitness values receive a larger number of copies. Crossover represents randomized exchange of parts of binary strings, whereas mutation randomly changes the values of bit positions.[2]

An important feature of the GA that contributes to its efficiency is *implicit parallelism*, a parallel processing of a large number of schemata. A *schema* [Holland (1975)] is a similarity template describing a subset of strings with similarities at certain string positions. They are described over a ternary alphabet, $\{0, 1, *\}$, where 0 and 1 are considered as specific bits and $*$ is a "don't care" symbol. Holland's schema theorem asserts that the number of schemata with above-average fitness increases exponentially over time. In addition, Holland (1975) demonstrated that the operation of reproduction and crossover on successive generations causes the candidate solutions to grow in the population approximately at a mathematically optimal rate. Optimality is defined by viewing the search strategy as a set of multi-armed bandit problems, for which the optimal strategy is known. In a multiarmed bandit problem, the agent has to choose each period between one of the n alternatives with constant expected payoffs, which are unknown a priori.

In the cobweb model, there are n firms in a competitive market that are price takers and that produce the same, perishable good. Because of a production lag, the quantity produced depends on the expected price level. The cost of a production of a firm i is given by

$$C_{i,t} = xq_{i,t} + \frac{1}{2}ny(q_{i,t})^2, \tag{1}$$

where $C_{i,t}$ is firm i's cost of production for sale at time t, $q_{i,t}$ is a quantity it produces for sale at time t, and x and y are parameters greater than zero. The profit of an individual firm, $\Pi_{i,t}$, is given by

$$\Pi_{i,t} = P_t q_{i,t} - C_{i,t}, \tag{2}$$

where P_t is the price of the good at time t. The optimal quantity produced by firm i at time t is given by

$$q_{i,t} = \frac{1}{yn}\left(P^e_{i,t} - x\right), \tag{3}$$

where $P^e_{i,t}$ is firm i's expectation of price P_t. The resulting market-clearing price is given using the demand equation

$$P_t = A - B\sum_{i=1}^{n} q_{i,t}. \tag{4}$$

In the unique rational-expectations equilibrium (REE), where $P^e_{i,t} = P^e_t = P_t$, quantity $q_{i,t} = q^*$ and price $P_t = P^*$ are constant. If firms have *naive* expectations, where the expected price is equal to the last period's price, the model converges

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

to the REE for the *cobweb-stable* case, that is, when $B/y < 1$, and diverges away for the *cobweb-unstable* case, that is, when $B/y > 1$.

In the genetic algorithm application, firms' one-period decision rules are represented by binary strings. A firm i, $i = 1 \ldots n$, makes a decision about its production for time t using a binary string of finite length ℓ, written over $\{0, 1\}$ alphabet. A binary string is first decoded into an integer value and then normalized to give the quantity $q_{i,t} \in [0, q_{\max}]$, where $q_{\max}$ is the maximum quantity that a firm can produce. The quantity $q_{i,t}$ represents firm i's production decision at time period t.

Once the quantities are determined, the market-clearing price P_t is computed using (4). This price is in turn used to compute firms' profits at time t. A profit that firm i earns determines the *fitness*, $\mu_{i,t}$ of firm i's decision rule.

The population of decision rules is then updated to create a population of rules that will be used at time $t + 1$. We discuss two versions of the GA that are implemented to update firms' decision rules. The first one, the basic GA, includes application of reproduction, crossover, and mutation. The second one, the enhanced GA, besides these three, includes the election operator.

Reproduction makes copies of individual binary strings. The criterion used in copying is the value of the fitness function. Binary strings with higher fitness value are assigned higher probability of contributing an offspring that undergoes further genetic operation. There are several different ways to perform this operator. The one used for the cobweb applications is called *proportionate selection*, where a probability that a binary string i receives a copy $C_{i,t}$ is given by

$$P(C_{i,t}) = \frac{\mu_{i,t}}{\sum_{i=1}^{n} \mu_{i,t}} \qquad i = 1 \ldots n. \tag{5}$$

The algorithmic form of the proportionate selection is like a biased roulette wheel, where each string is allocated a slot sized in proportion to its fitness. The number of spins of the wheel is equal to the number of strings in a population. Each spin yields a reproduction candidate. Once a string is selected, its exact copy is made. When n copies of strings are made (the number of strings in a population is kept constant), the reproduction is completed. These copies constitute a *mating pool*, which then undergoes application of other genetic operators.

Another commonly used reproduction operator is *tournament selection*. A copy of a string is selected in the following way: Two strings are selected randomly. The fitness values of the two selected strings are compared and the copy of the one with the higher fitness value is made. Strings that participated in the tournament are placed back into the original population and thus each one can be selected for the tournament again. These steps are repeated n times to create a population of n copies.

Crossover exchanges parts of pairs of randomly selected strings. It operates in two stages. In the first, two strings are selected from the mating pool at random. Then, in the second stage, a number k is selected, again, randomly from $(1 \ldots \ell - 1)$ and two new strings are formed by swapping the set of binary values to the right

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

of the position k. The total of $n/2$ (n is an even integer) pairs is selected and the crossover takes place on each pair with probability p_{cross}. An example of the crossover between two chromosomes for $\ell = 12$ and $k = 5$ is given below:

$$\begin{cases} 1 \quad 1 \quad 0 \quad 0 \quad 1 \,|\, 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \\ 0 \quad 0 \quad 0 \quad 1 \quad 0 \,|\, 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \end{cases}.$$

The resulting two strings are

$$\begin{cases} 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \\ 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \end{cases}.$$

Two strings that undergo crossover are recorded as two parents and the resulting strings as two offspring.

Mutation is the process of random change of the value of a position within a string. Each position has a small probability, p_{mut}, of being altered by mutation, independent of other positions. The role of this operator is to maintain population diversity.

The *election* operator tests newly generated offspring before they are permitted to become members of a new population. A *potential* fitness based on last period's price is computed for each offspring. Then, two parents and two corresponding offspring are ranked according to their fitness value, from the highest to the lowest, and the top two are taken as members of the new population of decision rules. In case of a tie between a parent and an offspring, an offspring becomes a member of the new population.

The application of the genetic operators on the members of population t results in a population of rules that are used at time period $t + 1$. The above-described steps are applied for T iterations. Initial population at time period 0 is randomly generated.

The whole process may be given the following economic interpretation: Reproduction works like the imitation of successful rivals. Binary strings of these firms have high fitness values and are copied by others. Strings with lower fitness values, which means worse production decisions and lower profits, get fewer copies (or none) in the next generation. Crossover and mutation are used to generate new ideas on how much to produce and offer for sale, recombining existing beliefs and generating new ones. If an election operator is included, the above interpretation may be modified in the following way: In each period, firms generate new production decisions using genetic operators. They compare the fitness 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.

Note that in this application of the genetic algorithm adaptation, individual firms do not use first-order conditions for decision making, as they do in the other learning algorithms previously studied in the context of the cobweb model. Simulations using different cobweb and genetic algorithm parameter values converged for both cobweb-stable-and-unstable cases. The enhanced genetic algorithm converged to the REE values, whereas the basic GA approached the equilibrium and then continued fluctuating around it because of the continuing effects of mutation.[3]

Arifovic also implements the *multiple-population* GA, where each firm is endowed with an entire population of strings. We can think of these strings in such a population as an agent's mutually competing ideas about the right behavior in a given environment. In each time period, only one string is selected as a string that determines agent's behavior. The probability of choosing a particular string is proportional to its performance under predefined conditions. Although an agent chooses only one string from among the whole collection, she evaluates ex post *all* of the alternative ideas. Thus, in the context of the cobweb model, in each time period t, a firm chooses one binary string from a whole collection and uses that as its actual production decision. Once the market-clearing price is computed, the firm uses that price to compute profits that each string in the collection would have earned at that price level. These profits determine the binary strings' fitness values. Once fitness values are computed, the application of GA operators takes place within each population of strings, that is, at the level of each individual firm.

This is the framework that is richer than the single-population framework in that the firms have a number of different ideas about possible production quantities. Even though this is a more complex framework, the computational requirements are identical to those of the single-population case. Application of the election operator is required for the convergence of the multiple-population GA to the REE. When the convergence occurs, all of the binary strings, in all of the populations, decode to the REE quantity. Without the election operator, simulations are characterized by wide and erratic fluctuations that do not die out over time. So, it appears that in a model of individual learning, agents have to be more sophisticated, that is, use the election operator, in order for the evolutionary model to converge to REE.

In the single-population framework, adaptation and learning take place at the level of the entire population. This type of learning can be referred to as *social learning*. On the other hand, in the multiple-population framework, learning takes place at the individual level and thus this type of learning can be referred to as *individual learning*. Note that we could add elements of social learning to the multiple-population framework by allowing individual agents to observe occasionally some of the strategies that other agents are contemplating or using.

The GA behavior and the patterns observed in the experiments with human subjects are also contrasted with the patterns generated by three other commonly used learning algorithms: cobweb expectations, sample average of past prices, and least squares. This represents an example of how we can develop the criteria for testing the performance of different learning algorithms and address the issue related to the arbitrariness of choice of a particular algorithm. The objective is to examine whether a model based on a particular set of behavioral assumptions consistently outperforms models based on different behavioral assumptions in capturing and explaining the behavior observed in the experiments with human subjects.

Three aspects of the Wellford's (1989) experimental data, namely the absence of divergent patterns in the cobweb-unstable case, fluctuations around cobweb model equilibrium values and the greater price variance of the unstable case are used for the evaluation of the performance of the above three learning algorithms and the GA. Arifovic shows that the GA exhibits the same behavior, that is, is able to capture all three features of the experimental data and does better than any of the other three algorithms. For example, the convergence of moving-average schemes is smooth without fluctuations, least-squares and cobweb expectations diverge away for the cobweb-unstable case. At the same time, GA (both single- and multiple-population designs) converge for both stable and unstable cases, exhibiting fluctuations around the equilibrium values along the transition path. Finally, GA price patterns also show that the price variance is greater in simulations of the unstable case than in the simulations of the stable case. The difference is statistically significant.

2.2. A Model with Fixed Costs and Entry/Exit Decisions

Dawid and Kopel (1998) use a version of the cobweb model with fixed costs to illustrate how the outcomes of genetic algorithm simulations can depend on the details of a particular coding scheme. In their version of the cobweb model, the costs are given by $C_{i,t} = z + yq_{i,t}^2$ if $q_{i,t} > 0$ and equal to 0 if $q_{i,t} = 0$. For a low enough fixed cost z, the homogenous REE exists where all firms produce identical quantities. As the size of z increases, there exists a heterogeneous equilibrium with a fraction of firms producing positive quantities, while the remaining firms stay out of the market. Both those that produce and those that do not earn zero profits in this equilibrium. Finally, when z reaches a high enough value, there is no equilibrium in which firms (or fraction of them) produce positive quantities. The implementation of the genetic algorithm is the same as the one described in the single-population design; that is, each firm's production decision is represented by a binary string and reproduction, crossover, and mutation are used for the updating of the decision rules.

For the set of the parameter values for which the unique, homogenous REE exists, the genetic algorithm population converges to the REE quantity. For the set of the parameter values for which only the heterogeneous equilibrium exists, the genetic algorithm population converges to the positive quantities that are optimal, given the observed price. However, fixed costs are so high that firms make negative profits. Thus, this is not a REE because each individual firm would be better off by exiting the market and making zero profits instead. Note that a homogeneous state in which all firms set their production decisions to zero is not a REE either. In that case, the market price would be so high that an individual firm would benefit by making a unilateral decision to start producing again. Dawid and Kopel use the above simulation result to demonstrate how the coding scheme can have an impact on the results of GA adaptation using the condition for local asymptotic stability for the GA dynamics.

A uniform state e_k (represented by a binary string) is locally asymptotically stable with $p_{\text{mut}} = 0$, and one-point crossover with probability $p_{\text{cross}} \in (0, 1]$ if

$$\frac{d(j,k)}{\ell - 1} > \frac{1}{p_{\text{cross}}}\left[1 - \frac{\mu_k(e_k)}{\mu_j(e_k)}\right] \tag{6}$$

for all $j \in \Omega$, where Ω is a set of all possible states that can be encoded with a string of length ℓ, $j \neq k$, e_k is a state under consideration, e_j is the state with a higher fitness value, $d(j, k)$ is the distance between the two outermost bits where j and k differ in value. If there is one $j \neq k$ such that the inequality holds the other way round, e_k is unstable.[4]

This condition shows that a state consisting almost only of strings k will converge to the uniform state e_k if the strings j, which are receiving a higher fitness in the current state, differ from k in bits positioned far apart. The number of strings j will grow in proportion to their fitness. However, as long as the ratio between $\mu_j(e_k)$ and $\mu_k(e_k)$ is not too large, there is relatively high probability that a string j will be paired with a string k and thus destroyed during crossover because any crossover point between the two outermost differing bits will destroy j.

Dawid and Kopel show that the state e_k in which all firms produce positive quantities and make negative profits is locally stable with respect to the string j that prescribes zero production. This string earns zero profits and thus has higher fitness value than string k with negative profits. The string k that encodes positive quantity (0.7143) is given by 1011011011, and the string that encodes zero production is a string of all zeros. Strings j and k differ in the first and last bits, and thus $d(j, k) = 9$. Dawid and Kopel show that, because the value of $d(j, k)$ divided by $\ell - 1$ outweighs the effects of the difference in the fitness values, inequality (6) holds, implying the local asymptotic stability of state e_k.

A modified coding scheme in which an extra bit is added to the binary string that is interpreted as an entry/exit decision (if it is equal to 0, stay out of the market and set $q_{i,t} = 0$) changes the stability results. The state that decodes to a positive production quantity loses its stability because the decision to switch between q^* and 0 requires a change of only a single bit, the entry/exit one.

Separation of the decision to enter or exit the market and a quantity determination leads to the behavior of the algorithm in which a fraction of the firms produce positive quantities while the remaining firms stay out of the market. The fraction corresponds to the one required for the existence of the heterogeneous REE.

2.3. Coevolution of Different Forecasting Rules

Another interesting extension of the GA cobweb model is a paper by Franke (1998). The paper studies the coevolution of four different types of rules used to determine the quantities produced. The first is the “fixed-production” rule (QF strategy) in which a binary string encodes a quantity that will be produced. The second one is the “adaptive expectations” rule (AE strategy) where a binary string encodes the value of the parameter α that controls the speed of adjustment. The third is the "regressions of order 1" rule (R1 strategy). With this rule, a binary string encodes the length of the sample period that is used to compute the least-squares estimate of the coefficient β. This coefficient is then used to form price expectations. The fourth is the "regressions of order 2" rule (R2 strategy). According to this rule, least-squares estimate takes only prices of every second period. A binary string encodes again the length of the sample period. Whereas the QF strategy yields a quantity to be produced, the other three rules generate forecasts that are then used in the first-order conditions to determine optimal quantities.

A GA population consists of binary strings that represent these four different types of rules. It is initialized randomly with equal fractions of all four types of rules. Reproduction is implemented on the entire population. However, crossover is implemented separately on a subpopulation of each type of rule. These different rules compete with each other during the genetic algorithm evolutionary process. The system's "collective memory" consists of the pools of four types of strategies from which once-extinct types of strategies can be reactivated.

Franke uses the set of Arifovic's (1994) parameter values for the cobweb-unstable case. What are the stability properties of these different rules in the case that each is adopted by all firms? The QF strategy results in convergence to a REE [Arifovic (1994)]. The AE strategy is stable for a sufficiently low value of α. Both $R1$- and $R2$-strategy rules are locally unstable, irrespective of the sample size, for the cobweb-unstable case.

Franke examines the behavior of the system in an environment with a deterministic demand curve and in a stochastic environment where there is a disturbance to the demand curve that follows an AR(1) process. In a deterministic environment, the GA converges to the REE in which all firms produce the same quantities, but use different rules to make these decisions. On average, 75% use the AE strategy whereas the QF strategy almost disappears from the population with only 0.01% of firms, on average, using it.

The stochastic environment is characterized by the coevolution of strategies. During coevolution, different rules get wiped out from the population and brought back in from the collective memory. All four types of strategies survive in the population. The largest percentage are still the AE-strategy binary strings, but the second highest percentage is represented by QF strategies. At the same time, R1 and R2 strategies take much smaller proportions of the adapted populations. During the coevolutionary process, the total quantity produced by the GA strategies remains close to the equilibrium quantity.

Competition of different rules and their coevolution results in stability of the cobweb model that cannot be achieved when either one of the three forecasting rules is adopted by all firms. The coevolution is characterized by continual change in rules' fitness values and thus by continual extinction of those rules whose fitness falls below the average and their reappearance (when called from the collective memory). Overall, the competition among different forecasting rules has a stabilizing effect on the economic environment.

3. EQUILIBRIUM SELECTION

In this section, we look at the use of the evolutionary algorithms as equilibrium selection devices in three types of environments with multiplicity of equilibria: overlapping generations economies with fiat money, growth model, and search model of money.

3.1. Overlapping Generations Economies with Fiat Money

Overlapping generations (OG) economies with fiat money usually possess multiple equilibrium paths and multiple stationary equilibria. These stationary equilibria have different stability properties under the rational expectations and under the adaptive dynamics [see, e.g., Lucas (1986), Marcet and Sargent (1989)]. The section begins with a description of the Arifovic (1995) application of the GA to the OG economy with two-period-lived agents and two types of monetary policy, constant money supply and constant deficit finance through seignorage. The economy consists of overlapping generations of two-period-lived agents. Each generation consists of an equal number, N, of agents. Every agent of generation t lives over two consecutive periods, t and $t+1$, and consumes $c_t(t)$ in the first period (youth) and $c_{t+1}(t)$ in the second period (old age). Agents have identical preferences and endowment patterns. When young, each agent is endowed with w^1 units of a perishable consumption good, and with w^2 units when old ($w^1 > w^2$). The amount of fiat money that government supplies at time t is given by $H(t)$.

Each young individual faces the following maximization problem:

$$\max c_t(t) c_t(t+1)$$
$$\text{s.t. } c_t^t \leq w^1 - \frac{m(t)}{p(t)}$$
$$c_{t+1}(t) \leq w^2 + \frac{m(t)}{p(t+1)},$$

where $m(t)$ represents the nominal money balances that an agent acquires in the first period and spends in the second period of his life and $p(t)$ is the nominal price level at time period t. The perfect-foresight dynamics for $p(t)$ are given by

$$p(t) = \frac{S(t)}{S(t-1)} p(t-1), \tag{7}$$

where $S(t)$ are aggregate savings of agents of generation t. If the government pursues a policy of constant money supply, $H(t) = H$ for all t, the difference equation (7) has the unique stationary, Pareto-optimal, equilibrium with valued fiat money and constant price level; that is, $p(t) = p^*$ for all t, where $p^* = 2H/2(w^1 - w^2)N$. This equilibrium is unstable under the perfect-foresight dynamics, and is attainable only if the initial price is equal to p^*. There is also a continuum of monetary equilibria indexed by the initial price level p_0 in the interval (p^*, ∞).

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

All of the equilibria with an initial price greater than p^* converge to the stationary equilibrium in which money has no value.

In case of the second policy, the government finances a constant deficit, G, through seignorage; that is, $G = [H(t) - H(t-1)]/p(t)$. With this policy, the model has two stationary equilibria with valued fiat money, a low-inflation stationary equilibrium, π_1^*, and a high-inflation stationary equilibrium, π_2^*. The low-inflation stationary equilibrium is Pareto superior. The high-inflation stationary equilibrium is the stable solution, being the attractor for a continuum of rational-expectations equilibrium paths, starting from $\pi_0 \in (\pi_1^*, w^1/w^2)$. [If the initial inflation rate $\pi(0)$ is equal to π_1^*, the system attains a low-inflation stationary equilibrium.] The stability conditions also imply that an increase in the deficit results in a decrease in the inflation rate of a stable stationary equilibrium.

In the GA OG economy, there are two populations of binary strings at each t. One represents the rules of the young, members of generation t, and the other the rules of the old, members of generation $t-1$. Each population is updated in alternating time periods, after its members have gone through a two-period life cycle. Genetic algorithm strings encode the values of first-period consumption.

A member i, $i \in \{1, \ldots N\}$, of generation t makes a decision about the first-period consumption at time t using a *binary string*. Savings of agent i of generation t, $s_{i,t}$, are given as $s_{i,t} = w^1 - c_{i,t}^1$.

The sequence of events that takes place at time t is the following: First-period consumption values are obtained from decoded and normalized binary strings and individual savings are computed. Next, the value of aggregate savings is obtained. Aggregate savings, together with money holdings of agents who were born at $t-1$, determine the price level of the good that prevails at t. The price of the consumption good at time t is given by

$$p(t) = \frac{H}{\sum_{i=1}^{n} s_i(t)} \qquad \textbf{(8)}$$

in the economy with constant money supply, and by

$$p(t) = \frac{\sum_i^N s_i(t-1)p(t-1)}{\sum_i^N s_i(t) - G} \qquad \textbf{(9)}$$

for $\sum_i^N s_i(t-1) > G$, in the economy with constant deficit.

Then, the second-period consumption by member i, $i \in \{1, \ldots, N\}$, of generation $t-1$ is determined:

$$c_{i,t}(t-1) = \frac{s_i(t-1)p(t-1)}{p(t)} + w^2. \qquad \textbf{(10)}$$

Finally, fitness values for the members of generation $t-1$ are computed. The fitness of a string i of generation $t-1$ is given by the value of agent i's utility at $t+1$ (the second period of life):

$$\mu_{i,t-1} = U_i(c_{i,t-1}(t-1), c_{i,t-1}(t)) = c_{i,t-1}(t-1)c_{i,t-1}(t). \qquad \textbf{(11)}$$

The population for generation $t+1$ is generated from the population of generation $t-1$, using the genetic operators of reproduction, crossover, mutation, and election. Once the population of new generation $t+1$ is created, the whole cycle is repeated. The population of rules of generation $t+1$ represents the young agents, whereas the members of population t become the old agents. The populations of generations 0 and 1 are randomly generated. The system starts off with Nh units of money distributed to the initially old.

Simulations of the environment with constant money supply converge to the stationary equilibrium in which fiat money is valued. This equilibrium is also the point of convergence of the adaptive algorithm that uses the sample average of past price levels for the price forecasting [Lucas (1986)]. Likewise, the experimental OG economies simulated by Lim et al. (1994) exhibited price paths close to the stationary monetary equilibrium.

In the economy with the positive value of deficit, the GA converged to the low-inflation stationary equilibrium. The least-squares learning algorithm studied by Marcet and Sargent (1989) also converges to the low-inflation stationary equilibrium. Results of simulations show that the GA also converges for deficit values and initial conditions for which least-squares exhibited divergent behavior.[5] Inflationary paths observed in experiments with human subjects [Marimon and Sunder (1993), Arifovic (1995)] converged to the neighborhood of the low-inflation stationary equilibrium. Moreover, the experimental economies did not exhibit divergent inflationary paths in cases of deficit values and initial conditions for which least-squares did not converge. Thus, the GA performs better in capturing the features of the experimental data.

Modeling the evolution of expectations of the inflation rate. A different approach to modeling agents' decision rules by the GA OG environment is given in Bullard and Duffy (1999). In their model, a binary string is used to encode expectations of the inflation factor in an environment with constant deficit financed via seignorage. At time t, an agent i of generation t is endowed with a string that encodes $b_i(t)$ that will be used as agent i's expectation of the inflation factor. Then, agent i's expectation of the nominal price level $p(t+1)$ is given by

$$F_i[p(t+1)] = b_i(t)p(t). \tag{12}$$

These forecasts are then used in the first-order conditions of agent's maximization problem to solve for the optimal value of savings given the expectations $b_i(t)$. The model resulted in the convergence to the low-inflation stationary equilibrium with π^l. Larger values of G required more iterations for convergence. Thus, this method of applying the genetic algorithm to the overlapping generations model results in the selection of the same equilibrium.

Notice that in this environment, unlike that of Arifovic's (1995) model, the assumption of utility maximization is maintained. Bullard and Duffy argue that in "learning how to optimize" GA implementation where consumption decisions evolve, one assumes (implicitly or explicitly) that all agents have the same view of the future, and that, given the set of commonly held expectations, the GA is used to assign a value of choice variable to the agents. On the other hand, with their method, agents are "learning how to forecast." They have heterogeneous forecasts and are not sure about the beliefs held by other agents. Because of the lack of information and multiplicity of equilibria, there is a problem of coordinating on a particular equilibrium path.

However, recent evidence from OG experiments in which human subjects were asked to make both a savings decision and a forecast of the next period's inflation rate shows that actual savings decisions differ significantly from the optimal savings decisions that would be implied by individual forecasts [Bernasconi and Kirchkamp (1999)]. The question of what seems to be a better way of simulating decision making process in models with boundedly rational agents, as well as in the experiments with human subjects, remains open.

3.2. An *n*-Period Model

Another application of GAs to the OG environment is the Bullard and Duffy (1998a) model in which agents live for $n \geq 2$ periods. It is again a pure endowment economy with fiat money being the only asset that agents can use to save between periods. Each generation of agents receives the same endowment pattern such that $w_1 > w_2 > \cdots > w_n > 0$. This implies that agents of all generations will be able to achieve their optimal consumption and savings decisions by holding fiat currency. The government finances a constant deficit via seignorage. This general n-period model also has low stationary perfect-foresight equilibria in which fiat currency has positive value, low-inflation stationary equilibrium, and high-inflation stationary equilibrium.

The GA is implemented in the following way: At each time period t, there are n populations of binary strings that represent decision rules of n generations of agents alive at t. Each generation consists of an equal number, N, of agents. The total population of agents at time t is thus equal to $n \times N$. A binary string is used for the construction of an autoregressive forecasting rule. It is interpreted in the following way: One bit determines whether lagged values of prices or first differences of prices are used in autoregression. The remaining bits specify which lagged values are used. Using a specified autoregression, the forecast of the next period's inflation factor is computed. The forecast is then used to compute the optimal savings decision. Unlike the two-period setup, agents repeatedly use their forecasting rules to make savings decisions in $n - 1$ periods of their lives.

In this n-period setup, a binary string's fitness is not set equal to the agent's utility as in the models that have been described so far. Instead, the fitness is determined by the rule's out-of-sample mean squared error (MSE). The first half of the available data from past GA iterations is used to estimate the autoregression coefficient. The second half is used to calculate the MSE of a forecasting rule. The fitness value of a rule is simply the inverse of the MSE. The fitness value is recalculated in every period in which the binary string is used.

Binary strings of members of all generations that make savings decisions are updated at each t. A set of rules of agents born at time t is generated using reproduction (tournament selection), crossover, mutation, and election. During the reproduction step, Bullard and Duffy implement *mixing*. With mixing, the entire population of $n \times N$ rules is used to select copies of binary strings. This way, the updating uses the entire stock of genetic information available at t.

Once the rules of agents born at t are determined, members of middle generations experiment with new ideas prior to making savings decisions at t. This process is called *emulation*. At each time t, a member of each middle generation meets another member from the entire $n \times N$ population. Two rules undergo crossover and mutation in order to create two alternative rules. At this point, a version of the election operator is implemented. The fitness values of newly generated rules are calculated, and four rules (two old ones and two new ones) are ranked according to their fitness values. The agent adopts a rule with the highest fitness value.

Once the forecasting rules for all agents who make savings decisions at t are determined, optimal savings are computed for each individual agent. Finally, aggregate savings and prices are computed. If the aggregate savings falls short of the amount required to finance the government deficit, a situation interpreted as a *currency collapse*, the government imposes a taxation scheme to satisfy the deficit financing constraint. (The tax that every individual has to pay is such that the deficit financing constraint is more than satisfied.)

Two kinds of outcomes were observed in the GA economies that were simulated for $n = 3$ and $n = 7$, and for three different values of G. The first outcome was convergence to the low-inflation stationary equilibrium. When the convergence occurred, forecasting rules could correctly predict the gross inflation factor even though the population consisted of heterogeneous rules. Thus, in this economy, the convergence to a stationary equilibrium need not imply identical forecasting rules across the individuals.

The second outcome was a convergence to autarchy—a state of persistent currency collapse. This outcome was observed more frequently for larger values of n. One way to interpret this result is that the frequency of the persistent currency collapse increases with the increases in the number of times that agents are allowed to change their forecasts. For a given n, currency collapse was more frequent for larger values of G. As the value of G gets closer to $G_{\max}$ (maximum value of deficit that can be financed given agents' lifetime endowment profiles) a larger number of the economies experience the currency collapse.

3.3. Cycles and Equilibrium Selection

The issue of equilibrium selection also arises in the OG economies that are, in addition to steady states, characterized by periodic equilibria. Arifovic (1998) and Bullard and Duffy (1998b) look at the stability of periodic equilibria under the GA dynamics.

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

Two-period cycle. We begin with a description of the GA dynamics [Arifovic (1998)] in the OG economy where, in addition to two steady states, autarky, and monetary steady state, a two-period cycle exists. Agents receive one unit of labor endowment when young. They work when young and consume when old. The production function is linear, and one unit of labor, n_t, yields one unit of output, y_t. Fiat money is the only means for saving. The stock of nominal money is constant over time; that is, $H(t) = H$ for all t. Agents' preferences are given by

$$\max ac_t(t+1) - \frac{1}{2}bc_t(t+1)^2 - n(t),$$

where $a > 0$, $b > 0$, and $n(t)$ is labor supply when young.

Market clearing requires $y(t) = n(t)$ and $n(t) = H/p(t)$. From the condition $p(t)n(t) = H$, it follows that a gross rate of return on money holdings, $R(t) = p(t)/p(t+1)$ is equal to $n(t+1)/n(t)$. Using first-order condition and market-clearing conditions, the equilibrium law of motion for the labor supply is given by

$$n_t = \frac{a(n_{t+1}/n_t) - 1}{b(n_{t+1}/n_t)^2}. \quad \textbf{(13)}$$

For the values of the parameters chosen for the GA application, the model has a unique monetary steady-state equilibrium and a two-period cycle. In the monetary steady state, $n_t = n^*$, and $p_t = p_{t+1} = p^*$ for all t. The rate of return on money $R(t)$, is also constant; that is, $R(t-1) = R(t) = R^*$. In the two-period cycle, the endogenous variables cycle between two sets of values. The first one $n^{*,1}$, $p^{1,*}$, and $R^{1,*}$ is associated with high labor supply, low nominal price level, and high rate of return on money. The other, $n^{*,2}$, $p^{2,*}$, and $R^{2,*}$ is associated with low labor supply, high nominal price level, and low rate of return on money.

The application of the GA is identical to the one described for the pure-endowment OG economy, except that now instead of first-period consumption, a binary string encodes a labor-supply decision. The equations that determine the values of savings and prices correspond to those given in Section 3.1 for the pure-endowment economy. Fitness values of binary strings are given as values of agents' utilities.

GA simulations always converged to the two-period cycle, even when initialized in the neighborhood of the monetary steady state. Analysis of the evolutionary dynamics shows, first, that if the GA converges, it converges to the stationary equilibrium values, and, second, that the GA dynamics select the two-period cycle over the monetary steady state. The two-period cycle is globally stable under the evolutionary dynamics.[6]

The dynamics are defined by the impact of wealth and intertemporal substitution effects on the evolution of decision rules. Suppose that the two GA populations are initialized in the neighborhood of the monetary steady state so that the average labor supply of the first population, $\bar{n}^{p1}(t)$, is smaller than n^*, whereas the average labor supply of the second population, $\bar{n}^{p2}(t)$, is greater than n^*. This means that the first population will enjoy the rate of return on money holdings $R^{p1}(t) > R^*$. The second

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

population will experience $R^{p2}(t) < R^*$. What is the impact of these rates of return on further updating of the decision rules? Because the wealth effect dominates the substitution effect in the neighborhood of the steady state, the relatively high $R^{p1}(t)$ will have negative effect on $\bar{n}^{p1}(t)$, and the relatively low $R^{p2}(t)$ will have negative effect on further evolution of $\bar{n}^{p2}(t)$. Thus, $\bar{n}^{p1}(t)$ decreases, and $\bar{n}^{p2}(t)$ increases, reinforcing the initial effects, that is, further increase in $R^{p1}(t)$ and further decrease in $R^{p2}(t)$. As this process continues, the labor supply values of the two populations will be driven away from the steady state and toward the two-period cycle.

If $\bar{n}^{p1}(t)$ and $\bar{n}^{p2}(t)$ overshoot the values of the two-period cycle, they will get into the region where the substitution effect dominates. The substitution effect will, however, halt further divergence of $\bar{n}^{p1}(t)$ and $\bar{n}^{p2}(t)$ and will bring them back toward the two-period cycle. This is because further increases in $R^{p1}(t)$ will now have a positive effect on $\bar{n}^{p1}(t)$, and further decreases of $R^{p2}(t)$ will now have a negative effect on $\bar{n}^{p2}(t)$.

The above analysis of evolutionary stability illustrates how the interaction between the GA dynamics that are based on the survival of well-performing rules and the underlying economic environment can be exploited to determine global stability properties of equilibria in the environments characterized by both steady states and periodic equilibria.

Higher-order cycles. Bullard and Duffy (1998b) examine GA learning in the Grandmont (1985) environment where, in addition to two steady states, and depending on the coefficients of relative risk aversion of the young and old agents, there are also periodic and chaotic equilibrium trajectories. It is again a two-period overlapping generations pure-endowment economy with preferences given by

$$U = \frac{c_t(t)^{1-\rho_1}}{1-\rho_1} + \frac{c_t(t+1)^{1-\rho_2}}{1-\rho_2},$$

where $\rho_1, \rho_2 \in (0, \infty)$ denote the coefficient of relative risk aversion of the young and old agents, respectively. The economy has two steady states: the Pareto-inferior autarchic steady state, and the Pareto-optimal monetary steady state. In addition, for the values of ρ_2 high enough and those of ρ_1 low enough, there are also periodic and chaotic stationary perfect-foresight equilibria. However, in these cases, the forward perfect-foresight dynamics are not well defined, but Grandmont (1985) showed that periodic equilibria of any order and chaos could exist as long-run outcomes in the backward perfect-foresight dynamics.

In the GA framework, each binary string decodes into an integer number, $k_i \in \{1, \bar{k}\}, i \in \{1, N\}$. The integer k_i prescribes the use of the price level that was realized $k+1$ periods in the past as the forecast of next period's price level. For example, consider a young agent i who has chosen k_i. Then, agent i's forecast of the price level at time $t+1$, $F_t^i[P(t+1)]$, is equal to $P(t-k_i-1)$. Based on this, the agent makes a forecast of the gross inflation factor and uses it to calculate the optimal savings decisions. This specification allows agents to adopt behavior consistent with steady-state or periodic trajectories for prices up to the limit $\bar{k}+1$. The length of a string was set to eight bits, implying that the agents could take actions consistent with a periodic equilibrium of an order as high as 256.

The results of simulations showed that GA agents could achieve coordination in this complicated environment, but they always coordinated on *simple* equilibria, steady state, or low-order cycle. For low values of ρ_2 (for which complicated stationary equilibrium trajectories do not exist), the GA agents always coordinated on the monetary steady state. For higher values of ρ_2, which imply multiplicity of stationary equilibria, the GA agents failed to coordinate on the periodic equilibria that are selected by the limiting backward perfect-foresight dynamics. The results of simulations showed that the GA agents could learn to coordinate on a monetary steady state and on periodic equilibria of order 2 and 3. The amount of time required for coordination increased with the value of ρ_2 and sometimes displayed qualitatively complicated dynamics for long periods of time prior to convergence to one of the low-order stationary equilibria.

This study demonstrated that (1) the introduction of GA learning does not imply a selection of a unique stationary equilibrium and (2) simple equilibria are more likely to be achieved. The first result is consistent with the results of studies of statistical learning algorithms.[7] These studies, which analyzed one-step forward-looking systems, showed that any stationary equilibrium can be locally stable. The results depend on the specified learning rule. For a k-period cycle, agents have to use k-order adaptive learning rules. Thus, the second GA result is new in that it reduces the set of "learnable" equilibria and suggests that low-order equilibria are more likely to be reached. Unlike in the previous studies, the GA environment allows for competition of forecasting rules of different order. In this competition, low-order rules dominate the simulations and lead to convergence to low-order periodic equilibria. It is worth pointing out that the result that simpler equilibria are more likely outcomes of learning is supported by the evidence from the experiments with human subjects.[8]

Because the behavior of the GA is governed by the performance of forecasting rules as measured by agents' utility, an interesting question is: What is it in the dynamics of the GA economy that drives these populations to low-order equilibria? This would involve the investigation of how forecasting rules that prescribe higher-order forecasts perform compared to the low-order rules. The dynamics of GAs are the result of interaction of the economic environment and the performance of different kinds of rules. This might provide some intuition for the results of simulations and better understanding of why simple equilibria seem to be more robust in the evolutionary sense than more complicated, high-order equilibria.

3.4. Growth and Development

Arifovic et al. (1997) study the adaptation of GA agents in a model of growth in which human capital accumulation is subject to increasing returns. The underlying economic model is a version of Azariadis and Drazen (1990) environment in which there are two steady states: low-income steady state (a poverty trap) and high-income steady state. Under the perfect-foresight dynamics, economies that start in the neighborhood of the low-income steady state always remain in the poverty trap. Arifovic and colleagues (1997) describe the mechanism through which the take-off can take place.

The model is an overlapping generations economy with a constant number of agents, N, born at each period t. Agents live for two periods and are endowed with one unit of time at every date t. All agents in this economy have the same preferences, $U = \ell\text{nc\_}\{i, t\}(t) + \ell\text{nc\_}\{i, t\}(t+1)$. There is a single, perishable good that is both consumed and used as an input into production. Output per unit of effective labor is produced according to a neoclassical production function, $f(k(t)) = k(t)^{\alpha}$ where $\alpha \in (0, 1)$ and $k(t)$ is the ratio of capital to effective labor. The rental rate on physical capital and the wage are given by $r(t) = \alpha k(t)^{\alpha-1}$ and $w(t) = (1-\alpha)k(t)^{\alpha}$, respectively.

A young agent i of generation t makes a decision whether to spend a fraction of time, $\tau_{i,t} \in [0, 1)$, in training. Each young agent inherits the level of efficiency units, $x(t)$, available in the economy at time t. The level $x(t)$ is the average of the efficiency units (accumulated human capital) of agents of generation $t-1$:

$$x(t) = \frac{1}{N}\sum_{j=1}^{N} x_{j,t-1}(t), \tag{14}$$

where $x_{j,t-1}(t)$ is the number of effective units of agent j of generation $t-1$ at time t. Young agents can combine this endowment $x(t)$ with a training decision $\tau_{i,t}(t)$ in order to receive $x_t^i(t+1)$ effective units of labor when they are old, through a common training technology denoted by $h(\tau_{i,t}(t), x(t))$. The key feature of the model is that the individual agent's return to training depends positively on $x(t)$. Thus, $x_{i,t}(t+1)$ is given by

$$x_{i,t}(t+1) = h(\tau_{i,t}(t), x(t))x(t) = 1 + \gamma(x(t))\tau_{i,t}(t), \tag{15}$$

and the private yield on human capital $\gamma(\cdot)$ is given by the sigmoid function

$$\gamma(x(t)) = \frac{\lambda}{1 + e^{-x(t)}} - \frac{\lambda}{2}, \tag{16}$$

which is strictly increasing in $x(t)$ and implies the bounds given by $\gamma(0) = 0$ and

$$\lim_{x(t)\to\infty} \gamma(x(t)) = \frac{\lambda}{2} \equiv \hat{\gamma}.$$

Parameter $\lambda > 0$ controls the returns to investing in human capital. The accumulation equation for $x(t)$ is then given by

$$x(t+1) = x(t)[1 + \gamma(x(t))\overline{\tau}(t)], \tag{17}$$

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

where $\overline{\tau} = 1/N \sum_{i=1}^{N} \tau_{i,t}(t)$. In addition, the agent also makes a decision about the fraction, $\phi_{i,t}$, of time t wealth, given by total wage earnings, that is saved. Thus, the savings are equal to

$$s_{i,t}(t) = \phi_{i,t}(t)w(t)(1 - \tau_{i,t}(t))x(t). \quad \textbf{(18)}$$

These decisions govern the accumulation of physical capital in this economy over time.

The model has two steady states, the low-income, poverty trap and the high-growth, high-income steady state. The first one is equivalent to the steady state of a neoclassical model with no growth in human capital and no technological progress. In this steady state, $\tau = 0$ for all i and all t and human capital remains at its initial level; that is, $x(t)$ must be constant for all t. The other is the steady state in which $\tau > 0$ for all i and all t. Thus, $x(t)$ is growing at the constant rate so that, for t large enough, $\gamma(t) \rightarrow \hat{\gamma}$. The low-income steady state is locally stable in the perfect-foresight dynamics, whereas the high-income steady state is saddlepath stable.

In the GA model, agent i's, $i \in \{1, \ldots, N\}$, decision about the fraction of time that is spent training, $\tau_{i,t} \in [0, 1]$, and his/her decision about the fraction of wealth that is saved, $\phi_{i,t} \in [0, 1]$, are represented by a binary string of length l, where $l/2$ bits are used for coding of each of the two decisions. The rules' fitness values are again equal to the values of the utilities earned at the end of the second period of life. The rules are updated using reproduction, crossover, and mutation. At each t, there are two populations of rules, one representing the young agent and the other representing the old ones.

Regardless of the initial conditions, evolutionary economies eventually take off and reach the high-growth steady state that is the globally stable equilibrium under these dynamics. Once the high-growth steady state is reached (which happens with probability 1), the economy stays there forever. The initial level of $x(t)$ is the variable crucial for the timing of the take-off. The lower the level, the longer the time that the economy spends in the poverty trap. Initially, the GA quickly reaches the low-income steady state. At that point, most of the decision rules prescribe no time investment in training because investment in human capital yields lower return and results in lower fitness values. However, because of the effects of mutation, there is always a small fraction of rules with positive values of $\tau_{i,t}$. Those rules disappear from the population because of selection pressure. Nevertheless, they contribute to the increase in $x(t)$. Over time, as $x(t)$ increases, it reaches the threshold level beyond which the return on human capital becomes high and fitness values of rules that invest in training increase. Once this happens, selection pressure changes because decision rules that call for investing positive amounts of time in training now yield higher fitness values than those decision rules that continue to instruct their owners to invest zero time in training. At this point, the GA quickly takes the economy toward the high-income steady state in which individual $\tau_{i,t}$'s take positive values. The transition phase is

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

relatively short, and once the economy is in the high-income steady state, it stays there forever.

Because the exact date of the take-off depends on the specific sequence of mutations that are responsible for the accumulation of human capital, economies that start with identical initial conditions can have different timing of the development take-off. In general, higher rates of experimentation with nonzero investment in training (i.e., higher rates of mutation) imply shorter average time required for the take-off.

The existing models of growth with multiple steady states do allow for sustained differences in growth rates across economies, but cannot explain how countries that are initially in poverty traps make the transition to a high-development steady state. At the same time, the evolutionary model is able to explain not just long periods of time spent in the neighborhood of poverty traps, but also eventual transition toward the high-growth steady states. It captures two important facts about development that most of the perfect-foresight models fail to take account of. The first is the result that for initially low levels of human capital per capita, levels that would characterize preindustrial economies, a population of boundedly rational agents spends many generations in the neighborhood of the low-income steady state before eventually taking off toward the high-income steady state. This accounts for the fact from Maddison (1982) that today's richest countries were once stagnant for hundreds of years. The second is the result that initially identical economies might experience take-off at very different times. This is important because different take-off dates imply very different levels of per capita income across economies once they all reach high-income steady state. This accounts for another development fact, that there is a large and persistent disparity in the levels of per capita income across nations [Parente and Prescott (1993)].

3.5. Search Model of Money

Marimon et al. (1989) study evolutionary adaptation in a search model of money [Kiyotaki-Wright (1989)]. Depending on the parameter values, this model can have different unique stationary equilibria, as well as multiple stationary equilibria. Using a version of Holland's (1975) *classifier systems*, Marimon et al. examine two issues: the learnability of different unique equilibria and the selection of equilibrium in an environment with multiple equilibria.

A classifier system is a collection of if-then rules called *classifiers* that have condition and action parts.[9] The condition part usually is defined over the trinary alphabet, {0, 1, #}, where the wild card # is interpreted as a "don't care" symbol. The wildcard character is an important part of the condition because it allows agents to decide which pieces of information are relevant and to ignore the others in an environment that is constantly changing. An action part can be a binary string or a string of real numbers. Each classifier has a strength that is the measure of its performance in a given environment.

In every t, a classifier that *posts* its message, thus determining the system's *action*, is selected using Holland's *bucket brigade* algorithm. First, a list of classifiers whose condition part matches the current state is determined. These classifiers submit the bids that are proportional to their strengths, and the classifier with the highest bid is chosen to post its message. This way, the rules that are highly fit are given preference over other rules. The bid is subtracted from the classifier's strength and paid to other classifiers responsible for generating the state to which the classifier was matched.

Then, the GA is used within the classifier system to generate new rules that are tried. Unlike the bucket brigade, which is implemented in every period t, the GA is applied after a given number of periods, T_{ga}. The call to the GA can be made randomly, with T_{ga} specifying the average value. It also can be done asynchronously across different classifier systems.

Only a fraction of the entire population of classifiers is replaced when the call to the GA is made. The classifiers that are replaced are chosen on the basis of poor performance. The invocation of GA learning also may be conditioned on particular events such as lack of match or poor performance. Finally, mutation is modified because classifier systems use the trinary alphabet. The mutated character changes to one of the other two with equal probability: $0 \to \{1, \#\}$, $1 \to \{0, \#\}$, $\# \to \{0, 1\}$.

In the Kiyotaki-Wright economy, there are three types of agents and three types of goods, $i = 1, 2, 3$. Type i receives utility from consuming good i, and can produce good $i^* = i + 1$ modulo 3. All goods are indivisible, and each can be stored at a cost s_k, $s_3 > s_2 > s_1 > 0$. All agents are assumed to receive the same utility from consumption $u(i) > s_3$ and $u(k) = 0$, $k \neq i$, and to have a commmon discount factor $\beta \in (0, 1)$. The economy lives forever and there is a large and equal number of agents of each type. The economy and each agent within it live forever.

In each time period, all agents are randomly matched without regard to their type. An agent has to make two decisions sequentially. The first is whether or not to trade, given the good she is holding and the good that the partner is holding. The trade occurs only if both partners propose to trade. The second is whether to consume a good that she is holding or to carry it to the following period. If the good is consumed, the agent automatically produces another good i^*. She is also charged the storage cost of either the good just produced or, if the decision was not to consume, of the good carried to the following period.

In the "fundamental equilibrium," type 1 stores only good 2 in order to exchange it with type 2. Type 3 stores only good 1 in order to exchange it with type 2. Type 2 stores good 1 half of the time, which is exchanged for good 3, and stores good 3 half of the time, which is exchanged for good 1. Good 1, with the lowest storage cost, serves the role of the medium of exchange.

In the "speculative equilibrium," in addition to good 1 being accepted by type 2 agents, good 3, which has the highest storage cost, becomes acceptable by type 1 agents. This speculative strategy is optimal whenever the difference in costs from storing good 3 rather than good 2 is less than the discounted expected utility benefit

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

of storing good 3 rather than good 2. Type 1 players are better off speculating in good 3 (rather than good 2) if storing good 3 makes it more likely that they will be able to successfully trade for their desired consumption good, good 1, and the additional likelihood of this event more than outweighs the additional cost of storing good 3 rather than good 2. This equilibrium is an example of the economy in which an endogenously determined medium of exchange, good 3, which has the highest storage costs, is dominated in the rate of return.

Depending on the parameter values, there are economies with a unique fundamental equilibrium, with a unique speculative equilibrium, as well as the economies where both of these stationary equilibria exist. The questions are whether these unique stationary equilibria can be learned by boundedly rational agents and what equilibrium is selected in the economies with multiple stationary equilibria.

In the Marimon et al. model, each agent uses two interconnected classifier systems. The first one, the *exchange* classifier system, determines the trading decision, and the second one, the *consumption* classifier system, determines the consumption decision. Agents care about long-run average level of utility.

The exchange classifier system consists of a list of strings of length 7. The first six bits are defined over the trinary alphabet and are used as a condition part, encoding the agent's own good stored and partner's good stored. Each of these takes three bits. For the trading decision, the code is in the binary alphabet $(1, 0)$, where 1 means trade, while 0 means don't trade. The consumption classifier consists of four positions. The first three positions are written over the trinary alphabet and encode the post-trade holdings of agent i. The fourth position can take a value of 1 or 0 and represents the action part that decides on whether or not the current holdings should be consumed. Here is an example of two exchange classifiers and their interpretation:

Own storage	Partner storage	Trading decision
1 0 0	0 0 1	1
0 0 1	# # 0	0

The first classifier instructs an agent carrying good 1 and who is matched with someone carrying good 3 to trade. The second classifier instructs an agent carrying good 3 and who is matched with someone not carrying good 3 not to trade.

The first three positions are written over the trinary alphabet and encode the post-trade holdings of agent i. The fourth position can take a value of 1 or 0 and represents the action part that decides on whether the current holdings should be consumed. Here is also an example of a consumption classifier that instructs the agent who is not holding good 1 (in a post-trade subperiod) not to consume:

Holdings	Consumption
0 # #	0

Once two agents are paired and the current holdings of goods are determined, exchange classifiers are matched to the current state. The exchange classifiers

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

whose condition parts are matched to the current state form a class of potential *bidders* that participate in an *auction*. The purpose is to select a classifier that makes the decision of agent i at time t. Among the matched classifiers, the rule with the highest strength is given the right to decide.[10]

The post-trade state of the world then is used to form a list of consumption classifiers whose condition parts match this state, and the consumption classifier from the list with the highest strength makes a consumption decision.[10]

Only the winning classifiers pay their bids by having them deducted from their strengths. The bid of the winning exchange classifier at t is paid to the winning consumption classifier at $t-1$, which is the classifier that is to be credited with setting the pretrading state at time t. The bid of the winning consumption classifier at t is paid to the winning exchange classifier at time t, which is to be credited with setting the postexchange state at t, thereby giving the winning consumption classifier a chance to bid. (If a classifier does not generate any immediate or future payoffs, it will drop out of competition because of the penalty of losing some of its strength when chosen.) In the Marimon et al. implementation, the updating of classifiers' strengths is formulated as a stochastic approximation algorithm where strengths represent cumulative averages of past payoffs.

In general, each agent in a model could have an exchange/consumption classifier system. However, in the Marimon et al. setup, all agents of the same type use the same classifier system to make their decisions.

Marimon et al. use several different operators to generate new rules. *Creation* is used in case there is no classifier that matches the condition part of the state. A new classifier is created with its condition part defined by the current state, and the action part randomly generated. *Diversification* is implemented in cases in which all classifiers whose condition part matches the message result in the same action, 1 or 0. A new classifier whose condition part encodes the current state and the action part opposite to that of the existing classifiers is added to the system.

Specialization is applied randomly with probability that is diminishing over time. The winning classifier is checked to see whether there are any #'s in the condition part. A new classifier is obtained by exposing each # to a probability of switching to either 0 or 1, depending on which one of the two encodes the current state that was matched to the classifier's condition.

A version of crossover called *generalization* is performed in two steps. First, two integers are drawn randomly, from [1, 7]. Then, with probability 0.5, the crossover is performed either *in*, meaning between the two numbers, or *out* to the left and to the right of the two selected integers. Within the chosen area, the specific value of any position (0 or 1) where the two classifiers disagree is changed to # for both strings. The children are each assigned strengths that are the average of their parents' strengths. For each child, a randomly selected individual from the population of "potential exterminants" is deleted.

For each set of the economic parameter values, Marimon et al. conducted simulations with (1) the complete set of rules where the GA was not implemented and (2) simulations with random initialization of classifiers and use of the genetic algorithm. For the economy in which only the fundamental stationary equilibrium exists, the classifier system converged to that equilibrium. However, for the economy in which the unique speculative equilibrium exists, the classifier system failed to converge to the equilibrium. Instead, it evolved toward the distribution of holdings that resembled the fundamental equilibrium. In the economy in which both the fundamental and the speculative equilibrium exist, the classifier system settled initially to the holdings quite similar to the speculative equilibrium holdings. However, it moved away from that equilibrium and settled in the fundamental equilibrium.

A number of other studies [Basci (1999), Staudinger (1998)] as well as the evidence from the experiments with human subjects [e.g., Duffy and Ochs (1999)] have shown that agents can coordinate on fundamental equilibria, but that they have difficulty coordinating on the speculative equilibrium. Basci simulates a system with a complete set of classifiers and introduces two modifications to induce players of type 1 to adopt the speculative strategy. First, he reduces the differential in storage cost between goods 3 and 2; second, he allows a certain fraction of agents to choose strategies according to their social (i.e., population-wide) values, which is referred to as *imitation*. Neither one of these, when implemented alone, helps the system to evolve toward the speculative equilibrium, but the combination of the two does result in a high frequency of convergence to the speculative equilibrium. Staudinger, who uses a GA implementation, also varied the size of the storage cost differential and found that, for small enough differential, the GA always converged to the speculative equilibrium. Overall, speculative equilibria seem to be difficult to learn both by boundedly rational agents as well as by human subjects. This is another example where, in addition to the equilibrium selection result, we can say that there are equilibria that are "hard" to learn. In this case, this is also supported by the evidence from the experiments with human subjects.

The fact that it is very difficult for both the boundedly rational agents and human subjects to adopt speculative trading strategies raises the question about the plausibility of the speculative equilibrium in the Kiyotaki-Wright environment and the question whether this represents a good way to model the rate-of-return dominance of money.

4. EVOLUTIONARY DYNAMICS IN THE ABSENCE OF EQUILIBRIUM BEHAVIOR

In this section, we look at several evolutionary models of bounded rationality that result in the long-run dynamics different from the rational-expectations dynamics. We borrow a term from Bullard (1994) and call these dynamics *learning equilibria*. The common feature of the models considered in this section is that there is persistence in fluctuations of the prices and excess volatility of the models' variables. These evolutionary economies do not converge to REE and exhibit the features of long-run behavior different from the REE dynamics.

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

4.1. Model of Mutual Fund Investors

Lettau (1997) uses GAs to model behavior of mutual fund investors in a simple portfolio-choice model. There is a single risky asset whose value, v, is normally distributed. The price of this asset, p_0, is constant and given exogenously to mutual fund investors. The agent's utility is given by $U(W) = -\exp(-\lambda W)$, where λ is the coefficient of absolute risk aversion and W is the net payoff equal to $(v - p_0)$. Under rational expectations, the optimal solution for a demand for the risky asset, s, is $s^* = \alpha_1^* \overline{v} - \alpha_2^* p_0$ with optimal coefficients α_1^* and α_2^*, $\alpha_1^* = \alpha_2^* = 1/\gamma\sigma_v^2$.

Mutual fund investors are represented by a population of binary strings. A binary string decodes into the value of $\alpha_{i,t}$ that investor i uses at time t. During a time period t, $t \in \{1, \ldots, T_{\max}\}$, there are S realizations of the asset's return, and investors make S portfolio decisions using their values of $\alpha_{i,t}$'s that remain fixed during t. After S realizations, the population is updated using reproduction, crossover, and mutation. The fitness of each rule is given as a cumulative utility over S drawings. Mutation decreases exponentially over time.

Lettau's simulations show that GA adaptive behavior leads to risk-taking bias that depends on the number of market observations S that the agents use before they update their investment portfolio. For small values of S, GA converges to $\overline{\alpha} > \alpha^*$ indicating overinvestment into the risky asset. This bias vanishes as S gets large. Because of the distribution of asset returns, the selection pressure promotes lucky rules that overinvest in the risky asset for small S. As S increases, agents are exposed to more trials and the chances of doing well due to luck decrease.

However, the patterns of behavior of mutual fund investors are much more volatile than what can be captured by a simple optimal portfolio-choice model or an adaptive model that converges to a constant portfolio-choice value. In fact, Lettau's analysis of the data on the flows in and out of mutual funds reveals that flows into mutual funds are positively correlated with returns, that flows are more sensitive to negative returns than to positive ones, and that evidence is stronger for riskier mutual funds.

To capture this more volatile behavior, Lettau modifies his GA model by introducing entry and exit of investors. At each t, a fixed number of new investors whose investment strategies are randomly generated enter the population and replace the same number of existing investors. The investors (or strategies) that exit the population are randomly determined. In addition, he sets $S = 1$, and keeps the rate of mutation constant for the entire length of each simulation. The continuous source of diversity from exit and entry, the minimal value of S, and a nondecreasing mutation rate resulted in persistent volatility of the value of $\overline{\alpha}$.

Lettau conducts two sets of GA simulations. (1) using the normal distribution of returns with the mean and variance values equal to those calculated from the actual mutual fund data, and (2) using the actual mutual fund returns. In both sets of simulations, GA agents exhibit the same pattern of behavior that looks very much like the behavior of mutual fund investors; that is, flows into mutual funds are positively correlated with returns, and flows are more sensitive to negative returns than to positive ones. As with the patterns of actual mutual fund flows, the GA portfolio adjustment is more extreme for high-risk mutual fund returns.

4.2. Artificial Stock Market

The underlying economic environment of the artificial stock market [Arthur et al. (1997), LeBaron et al. (1999)] is again the simple asset-pricing model where two assets are traded. The first is a risk-free bond which is in infinite supply and pays a constant interest rate, r. The second is a risky stock, in fixed supply N, that pays a stochastic dividend, d_t, which follows an autoregressive process. The model has a richer structure than the Lettau's model. Prices are determined endogenously on the basis of agents' demands for risky assets and the fixed supply of the asset. The functional form of the agent's utility is again constant absolute risk aversion (CARA) utility given by $U(W) = -\exp(-\lambda W)$. Under CARA utility and Gaussian distributions of risky asset prices, agent i's demand, $x_{i,t}$, for holding shares of the risky asset is given by

$$x_{i,t} = \frac{E_{i,t}(p_{t+1} + d_{t+1}) - p_t(1+r)}{\lambda\sigma^2_{i,t,p+d}}, \qquad \textbf{(19)}$$

where p_t is the price of the risky asset at t, and $\sigma^2_{i,t,p+d}$ is agent i's forecast of the conditional variance of $p + d$. (This relationship holds in the linear rational expectations equilibrium where distribution of prices is Gaussian.) Given these individual demands, the equilibrium price p_t can be computed from the market-clearing condition where total demand must equal the number of shares issued; that is,

$$\sum_{i=1}^{N} x_{i,t} = N. \qquad \textbf{(20)}$$

In the homogenous rational expectations equilibrium, the optimal forecasting function has the following form

$$E(p_{t+1} + d_{t+1}) = a(p_t + d_t) + b, \qquad \textbf{(21)}$$

where $a = \rho$ and b is a constant term that includes the values of $\rho, \bar{d}, r, \lambda$, and σ^2_{p+d}.

In the artificial stock market setup, agents' subjective expectational modes are represented by sets of rules that map states into forecasts. Each agent, $i \in \{1, \ldots, N\}$, is endowed with a population of M condition/forecast rules that are a modification of Holland's "condition-action" classifier system. The set of states includes both "technical" information that compares current price to moving averages of different length (important for detecting a trend in prices), and "fundamental" that includes information on dividend price ratios.

A condition part of a rule is defined over a trinary alphabet, of 0, 1, and # (don't care), and consists of 12 bits corresponding to states. A forecast part of the rule is a real-valued vector of length 3 corresponding to linear forecast parameters, and a

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

conditional variance estimate. A forecast of a given rule j of agent i is then given as a linear combination of the price and dividend by

$$E_{i,j}(p_{t+1} + d_{t+1}) = a_{i,j}(p_t + d_t) + b_{i,j} \tag{22}$$

with $\hat{\sigma}^2_{p+d} = \sigma^2_{i,j}$.

At time period t, an investor i makes a decision about the demand for risky asset $x_{i,t}$ in the following way: First, the list of rules whose condition part matches the current state is determined and the most accurate rule is chosen. The investor uses the forecast estimates of this rule to formulate the demand for shares by substituting the values $a_{i,j}$, $b_{i,j}$, and $\sigma^2_{i,j}$ into equation (22). Once all of the demands are submitted, the current price p_t is calculated using (19) and (20). At the end of the trading period t, agents update the accuracy of all the matched forecasting rules according to an exponentially weighted average of squared forecast error,

$$v^2_{t,i,j} = \left(1 - \frac{1}{\tau}\right) v^2_{t-1,i,j} + \frac{1}{\tau}\{(p_t + d_t) - [a_{i,j}(p_{t-1} + d_{t-1}) + b_{i,j}]\}^2. \tag{23}$$

The GA is applied on average every k periods, asynchronously across the agents. A fraction of worst-performing rules is replaced by new rules generated through tournament selection, *uniform* crossover, and mutation. With uniform crossover, the new child's bit string is built one bit at a time, choosing a bit from each parent in the corresponding position with equal probability. For the real-number part of the rule, three different methods of crossover are used at random. First, all of the real values are chosen from one parent selected at random. Second, a real value is chosen from each parent with equal probability. Third, the new values are created from a weighted average of the two parents' values, using $1/\sigma^2_j$ as the weight for each rule. The weights are normalized to sum to 1.

Two types of mutation are performed. For the condition part, the bits are flipped at random. For the forecast part, either new values are chosen randomly from the allowable ranges, using a uniform distribution, or the values are changed by a small, randomly chosen, amount.

Two sets of simulations were conducted. One in which the GA was implemented every 1,000 periods on average, and the other in which it was implemented every 250 periods on average, called slow-, and fast-learning cases, respectively. The two sets of simulations gave qualitatively different patterns of behavior. In the slow-learning simulations, the market price quickly converged to the rational expectations price, whereas in the fast-learning simulations the *complex* behavior emerged.

LeBaron et al. (1999) examine the time-series properties of the asset prices generated in two sets of simulations in greater detail. The fast-learning simulations show evidence for the presence of ARCH effects that is also a feature of actual asset-price time series. Both slow- and fast-learning cases show little indication of autocorrelation in the residuals of asset returns, which is similar to actual markets. Finally, both cases show persistence in trading volume series and positive contemporaneous correlation between volume and volatility, again the feature of actual series.

Tests also showed additional predictability coming from technical trading bits as well as fundamental bits for the fast-learning case, while no such extra predictability was found in the slow-learning case. Both technical trading rules [see Brock et al. (1992)] and dividend price ratios [see Campbell and Shiller (1988)] have been shown to have some predictive value with actual time series of asset prices. Note that in the rational-expectations equilibrium, neither one of these two types of indicators, technical or fundamental, reveals any extra information because all of the relevant information is contained in the lagged price.

Classifier systems in economic environments. We have seen two applications of the classifier systems to economic environments, to the search model of money and to the mean-variance asset-pricing model. Classifier systems are designed to evolve a set of rules that work well in the environments that require the use of different rules, avoiding the undue effects of competition among them. It is worth pointing out that an equivalent GA can be constructed for a number of environments where different parts of a binary string are used under different circumstances. The Kiyotaki–Wright environment is one such example. As we have seen, Staudinger (1998) used a GA representation to evolve the set of rules for this environment. Classifier systems can be interpreted as models of individual learning, in some respects similar to the multiple-population GA. Again, elements of social learning can be added by having agents exchange some of their rules.

Both applications of classifier systems that have been described in the survey use the GA for the updating of rules. However, some of the applications do not use the GA and the updating of classifiers takes place through the bucket brigade algorithm only [e.g., Lettau and Uhlig (1999)]. In this case, the system is based on the reinforcement of the rules that are initially specified as population members whose asymptotic behavior can be characterized analytically.

Lettau and Uhlig (1999) implement this type of classifier system in a dynamic programming problem. Using stochastic approximation methods, they study the asymptotic outcome of classifier system learning and show that certain aspects of the classifier system are closely related to the value function in dynamic programming. In general, the learnable decision function is not unique and may not coincide with the optimal decision function even if that function is attainable. They use this result to explain the empirical puzzle of excess sensitivity of consumption to transitory income.

4.3. Search for Optimal Trading Rules

Another strand of literature has used evolutionary algorithms, genetic programming (GP) in particular, to search for optimal trading rules [Allen and Karjalainen (1999), Neely et al. (1997)] in stock and foreign exchange markets. This research does not strictly fall into the category of models of Section 4. However, we take

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

a detour in order to describe GP and its applications to finance because technical trading emerged as an endogenous phenomenon in the artificial stock market.

GP [Koza (1992)] is an extension of GAs where rules are represented as hierarchical compositions of functions of varying length. In these tree-like structures, the successors of each node provide the arguments for the function identified with the node. The terminal nodes (i.e., nodes with no successors) correspond to the input data. The set of functions appropriate to the particular problem is prespecified. Working with structures of varying length is intended to overcome the limitations of the fixed-length genetic algorithm strings.

Like GAs, GP also maintains a population of genetic structures. The evolution takes place in a way very similar to the GA, using reproduction, crossover, and mutation on a population of tree structures. Crossover recombines two tree structures by replacing a randomly selected subtree in the first parent with a subtree from the second parent. (The operation is subject to the restriction that the resulting tree must be a well-defined rule.) Mutations are introduced by using a randomly generated tree in place of the second parent with a small probability. The initial population of trees is randomly generated.

In the above applications, trading rules are represented as trees. Functions that are usually specified are real-valued functions that include a function that computes a moving average of past prices (average) in a time window specified by a real-valued argument (rounded to an integer when the rule is evaluated), arithmetic operations ($+, -, /, *$), a function returning the absolute value of the difference between two real numbers (norm), Boolean functions that include logical functions (if-then-else, and, or, not) and comparisons of two real numbers ($>, <$), Boolean constants (true, false), and real constants.

Allen and Karjalainen (1999) were the first to use GP to find profitable trading rules for the S&P 500 Index using daily prices from 1928 to 1995. They did not find evidence that GP rules could earn consistent excess returns (after transaction costs) over a simple buy-and-hold strategy in the out-of-sample test periods.

However, a search for well-performing technical trading rules in foreign exchange markets resulted in different findings. Neely et al. (1997) use it in the foreign exchange markets [four currencies (deutschemark, yen, pound sterling, and Swiss frank) against dollars, and deutschemark/yen, and pound sterling/Swiss frank) and find rules that generate economically significant excess returns after transaction costs when tested out-of-sample. It is interesting that the rules identified by GP are similar to those commonly used by technical traders.

GP has greater flexibility than GAs because it works with strings of variable length. However, trees that evolve through GP simulations often have a very complicated nested structure with a large number of levels and nodes, and sometimes a large number of redundant parts. Consequently, it is very difficult to interpret them. Although GP seems quite suitable for applications like the ones described above, where the task is to search for the rules from the available time series, they may be less useful for modeling learning in general-equilibrium-type models where learning interacts with the environment. Inability to determine what the agents'

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

rules actually mean and how they affect other endogenous variables hinders the usefulness of GP in this type of environments.

4.4. Model with Costly Rational-Expectations Predictor

Brock and Hommes (1998a) find that evolving the selection of prediction strategies can result in chaotic dynamics. They consider a version of the cobweb model in which a continuum of agents can choose between two different predictors, H_1 and H_2, that correspond to two types of expectations, *naive* (price this period will be just equal to the last period's price) and rational expectations. The fractions $n_{1,t}$ and $n_{2,t}$ of agents using H_1 and H_2, respectively, changes over time.

Agents use a discrete choice model along the lines of Manski and McFadden (1981) to choose a predictor that they will use in the following period. The fractions of chosen predictors change over time according to the profits earned by agents. Agents can buy a rational-expectations predictor, H_1, at small but positive information cost C. This indicates an extra effort that must be invested to obtain a more sophisticated price forecast. Alternatively, they can obtain a simple predictor, H_2, for free. At time period $t+1$, the market price p_{t+1} is given by

$$p_t = A - B(n_{1,t}q_{1,t} + n_2 q_{2,t}) \tag{24}$$

where $q_{1,t}$ and $q_{2,t}$ are optimal quantities chosen by agents using predictors H_1 and H_2, respectively, and

$$\pi_{1,t} = \frac{b}{2}p_t^2 - C, \qquad \pi_{2,t} = \frac{b}{2}p_t(2p_t - p_{t-1}) \tag{25}$$

are the profits realized at price p_t. The updated fraction of agents that will use predictor H_j, $j \in \{1, 2\}$, at time $t+1$, is given by

$$n_{j,t+1} = \exp[\beta\pi_{j,t}]/Z_t, \qquad Z_t = \sum_{j=1}^{2} \exp[-\beta\pi_{j,t}] \tag{26}$$

The parameter β is the *intensity of choice*, measuring how fast agents switch predictors, that is, how sensitive the mass of traders is to differences in fitness across trading strategies. The special limiting case with $\beta = +\infty$ corresponds to the neoclassical deterministic choice model, where in each period all agents choose the optimal predictor. If $\beta = 0$, the mass of traders distributes itself evenly between the two predictors. Equations (24)–(26) describe what Brock and Hommes call *the adaptive rational equilibrium dynamics*.[11]

The case particularly interesting for the dynamics of the system is again the cobweb-unstable case where the ratio of the slopes of demand and supply is greater than 1. If all agents use the simple predictor, the steady state with price p^* is unstable. If $C = 0$ and all agents employ rational expectations, the steady state is globally stable.

What happens when agents can use both predictors and $C > 0$? Consider first an initial state where prices are close to the steady-state value and almost all agents use naive expectations. The use of naive expectations results in a divergence from the steady state. As prices diverge from the steady-state value, the prediction error from the naive expectations will increase. Consequently, a number of agents willing to use rational expectations increases. As this fraction increases, prices eventually will be pushed back toward their steady-state value. At this point, the prediction error of naive expectations becomes small again. This implies that the net profit corresponding to the sophisticated predictor becomes negative because of the information costs and the whole process is repeated. The speed at which the switching occurs depends on the intensity of choice, β. The value of β determines dynamics of the system.

For the cobweb-unstable case ($B/b > 1$) and $C > 0$, the system is globally unstable for a large enough value of β. Brock and Hommes prove that there is a range of values of β for which the two-period cycle is stable. As β increases further, numerical simulations show that the two-cycle loses stability and two stable four-cycles are created, and for even larger values, simulations indicate the occurrence of a chaotic attractor. Brock and Hommes prove that the system is chaotic for a positive Lebesgue measure set of (high) β-values.[12]

Brock and Hommes point out that, with the analytical results they provide, their work can be viewed as complementary to the more numerically oriented evolutionary models of expectations [e.g., Arthur et al. (1997)]. In fact, there is a parallel between the intensity of choice that indicates how fast agents change predictors and the speed at which agents update their rules in artificial stock markets. Higher intensity of choice and more frequent updating of rules result in more complicated dynamics.

4.5. Model of Exchange-Rate Behavior

This part of the survey reviews the role played by evolutionary algorithms in modeling of the exchange-rate dynamics. Persistent fluctuations have characterized the behavior of the exchange rates ever since the flexible exchange-rate system was introduced. However, neither structural models nor time-series models have been successful in capturing a high percentage of the variation in the exchange rate at short- or medium-term frequencies [Meese and Rogoff (1983), Frankel and Rose (1995)]. On the other hand, evolutionary dynamics [Arifovic (1996), Arifovic (1999)] result in persistent exchange-rate fluctuations that are driven by changes in agent's beliefs.

Arifovic (1996) uses the GA in a two-country overlapping generations model identical to the one described in Section 3, except that now there are two currencies that are perfect substitutes. Agents can hold any of the two currencies without any restrictions. There are now two monetary authorities, one supplying $H_1(t) = H_1$ of currency 1 and the other $H_2(t) = H_2$ of currency 2. The exchange rate $e(t)$ between the two currencies is defined as $e(t) = p_1(t)/p_2(t)$, where $p_1(t)$ is the

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

nominal price in terms of currency 1 and $p_2(t)$ is the nominal price in terms of currency 2. When there is no uncertainty, the return on the two currencies must be equal,

$$R_1(t) = R_2(t) = \frac{p_1(t)}{p_1(t+1)} = \frac{p_2(t)}{p_2(t+1)}, \qquad t \geq 1, \tag{27}$$

where $R_1(t)$ and $R_1(t)$ are the gross real rate of return between t and $t+1$.

From equation (27) it follows that the exchange rate is constant over time:

$$e(t+1) = e(t) = e, \qquad t \geq 1.$$

Agents' savings, $s(t)$, in the first period of life, are equal to the sum of real holdings of currency 1 and currency 2. Aggregate savings that represent real-world money demand are equal to the sum of young agents' savings; that is, $S(t) = Ns(t)$. Because the rates of return on the two currencies are identical, the agents are actually indifferent as to which currency they hold. Because of this, equations for individual money demands are not well defined. This fact results in the indeterminacy of the exchange rate [Kareken and Wallace (1981)] that asserts that if there is a monetary equilibrium where savings demand and money supplies are equal for an exchange rate, e, then there exists an equilibrium for any exchange rate $\hat{e} \in (0, \infty)$, $\hat{e} \neq e$. Consider the equilibrium condition that aggregate savings (real-world money demand) equals real-world money supply:

$$S(t) = \frac{H_1 + H_2 e}{p(t)}. \tag{28}$$

Then, if there is an equilibrium for a price sequence, $\{p_1(t), p_2(t)\}$, for the exchange rate e, we can find a sequence $\{\hat{p}_1(t), \hat{p}_2(t)\}$ for the exchange rate $\hat{e}$ that results in the same sequence of real rates of return as the original price sequence and, in turn, in the same values of aggregate savings. The reason why this can be accomplished is the equivalence between the two currencies as savings instruments.

The GA economy has a structure identical to the one described before; that is at each t, there are two populations of binary strings, representing the rules of the young and the old agents. The difference is that now a binary string i that characterizes agent i of generation t consists of two parts. The first part represents the agents' savings decision and decodes into the number $s_i(t) \in [0, w^1]$. The second part represents the agent's portfolio decision, that is, what fraction of the savings to place in currency 1, and decodes into the number $\lambda_i(t) \in [0, 1]$. Agent i of generation t places the amount of $\lambda_i(t)s_i(t)$ into currency 1, and the remaining part given by $[1 - \lambda_i(t)]s_i(t)$ in currency 2.

The sequence of events corresponds to the one for the single-currency model. First, the nominal prices at time t and the rates of return between $t-1$ and t are computed. The exchange rate is determined. Next, the nominal holdings of currencies 1 and 2 are computed for each member of generation t. These holdings of monies are carried over to period $t+1$.

Agents of generation $t-1$ use their holdings of monies to purchase the consumption good. Their second-period consumption is computed and, finally, utilities of the members of generation $t-1$ are computed, which in turn determine the fitness values of their decision rules. At the end of a cycle of two periods, agents update their rules using reproduction, crossover, mutation, and election.

The results of simulations show persistent fluctuations in the exchange rate, whereas average values of savings stay close to the values of the stationary REE. The results are robust to the changes in the parameter values. Moreover, the stationary REE is unstable under the GA dynamics.

The out-of-equilibrium heterogeneity of the portfolio-fraction values results in the inequality of the rates of return on two currencies, $R_1(t)$ and $R_2(t)$. The rates of return depend on the movements of $s_i(t)s$ and $\lambda_i(t)s$ that belong to different populations, but since there is not much movement in $s_i(t)s$, the movements in $\lambda_i(t)s$ are driving the dynamics. Let us assume that savings remain at the stationary value, s^*, and that $H_1 = H_2$. Then, the rates of return on two currencies are given by $R_1(t) = \overline{\lambda}(t)/\overline{\lambda}(t-1)$, and $R_2(t) = [1-\overline{\lambda}(t)]/[1-\overline{\lambda}(t-1)]$. The GA agents seek to exploit this arbitrage opportunity by placing larger fractions of their savings into the currency that had a higher rate of return in the previous period.

In general, increasing sequences of $\overline{\lambda}(t)$ are required to preserve $R_1(t) > R_2(t)$, and decreasing sequences to preserve $R_1(t) < R_2(t)$. Suppose that $R_1(t-1) > R_2(t-1)$. At t, agents will attempt to put larger fractions into currency 1. Because of the election, $\overline{\lambda}(t) \geq \overline{\lambda}(t-2)$ (these are the λ's that belong to the same population of rules). However, it is the relationship between $\overline{\lambda}(t)$ and $\overline{\lambda}(t-1)$, which belong to different populations, that determines $R_1(t)$ and $R_2(t)$.

If $\overline{\lambda}(t) > \overline{\lambda}(t-1)$, the direction of the inequality is preserved; that is, $R_1(t) > R_2(t)$, the value of currency 1 increases. On the other hand, if the aggregate change is not large enough, $\overline{\lambda}(t) < \overline{\lambda}(t-1)$, and $R_1(t) < R_2(t)$. The reversal of inequality will prompt the agents to start placing more savings into currency 2, that is, start decreasing individual $\lambda_i(t)$'s. When this happens, $\overline{\lambda}(t)$ starts moving in the other direction. These dynamics bring about fluctuations in the portfolio fraction and, consequently, in the exchange rate that persist over time. Overall, very high and very low values of $\overline{\lambda}(t)$ are less likely than the intermediate values. As $\overline{\lambda}(t)$ takes on relatively high (low) values, the probability of obtaining higher (lower) values decreases. To obtain an increasing sequence of $\overline{\lambda}(t)$ values, more and more bits have to be switched from 0 to 1. As $\overline{\lambda}(t)$ increases, the number of these bits decreases. Thus as $\overline{\lambda}(t)$ gets larger, the probability of further increases will decrease. A similar argument applies for a decreasing sequence of $\overline{\lambda}(t)$ values and an increasing number of zeroes.

This prediction is confirmed by the results of simulations that show that the mass of distribution of the values of $\overline{\lambda}(t)$ is concentrated in the interval [0.4, 0.6]. The same distribution of the values of $\overline{\lambda}(t)$ was observed in the experiments with human subjects, where the same economies were simulated. Most of the values (87%) were concentrated in the same interval. Two other features of the experimental data are worth pointing out. The values of the individual portfolio fractions and thus the values of the exchange rates did not settle down, but kept fluctuating. At the same time, the values of the first-period consumption settled close to the stationary values. Thus the genetic algorithm simulations capture these features of the experimental data.

Arifovic (1999) studies the same model but uses a real number representation of the decision rules. Thus a decision rule of agent i of generation t is defined by two real numbers, $s_i(t) \in [0, w^1]$ and $\lambda_i(t) \in [0, 1]$. The updating takes place using imitation, experimentation, and election. The imitation operator in this setup is equivalent to the reproduction with proportionate selection. Experimentation is equivalent to mutation and is performed in the following way: A random number is drawn from the uniform distribution, for $s_i(t)$ from the interval $[0, w^1]$, and for $\lambda_i(t)$ from the interval [0, 1]. With this rule representation and updating scheme, the evolutionary algorithm represents a version of the stochastic replicator dynamics.[13]

The instability of a stationary equilibrium is preserved under this updating scheme and the main features of the dynamics remain the same. In particular, the mass of distribution of $\overline{\lambda}(t)$ is concentrated in the (same) interval [0.4, 0.6]. This represents one of the first studies that examine the dynamics under binary and real-number representation and demonstrate that the qualitative features of the dynamics are preserved under different rule representation schemes.

Overall, evolutionary models of the exchange rate generate persistent fluctuations of the exchange rates that result from agents' portfolio adjustments. It is worth pointing out that statistical learning algorithms might not generate the persistent exchange-rate volatility in the same OG environment. For example, the stochastic approximation algorithm converges to a stationary REE with the constant value of the exchange rate [see Sargent (1993)].

Rational agent versus boundedly rational agents. In the same exchange-rate model with real-number representation of decision rules, Arifovic (1999) introduces a rational agent who has enough knowledge of the model to be able to make two-period-ahead forecasts of the rates of return on two currencies. One rational agent is born at each t and lives for two periods. On the basis of these forecasts, the agent makes her optimal portfolio decision. Rational agents' decisions do not affect the price levels.

The performance of the rational agent, in terms of the average utilities earned over a long period of time, are compared to the performance of boundedly rational agents. The results show that the rational agent performs only slightly better than the average and the median boundedly rational agents. The differences are greater for larger rates of mutation, indicating that boundedly rational agents do worse if they experiment more. However, the best within a generation of boundedly rational agents always does better than the rational agent and the difference increases with the rate of experimentation.

Model with deficits. Finally, it is worth pointing out that study of the same type of the overlapping generations environment where two countries finance deficits

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

of different sizes through seignorage does not generate persistent exchange-rate volatility (Arifovic, in press). Note that this economy is also characterized by the indeterminacy of the equilibrium exchange rate.

The results show that the currency used to finance the larger of the two deficits cannot survive in a free competition between the two currencies. Evolution of agents' decision rules results in a flight away from this currency until it eventually becomes valueless. At the end of the adjustment process, agents hold all of their savings in the currency used to finance the lower of the two deficits. Thus, the economy converges to the equilibrium in which only the low-deficit currency is valued. This equilibrium is equivalent to a stationary equilibrium of the single-currency model. The speed of adjustment depends on the size of the difference between the two deficits. The larger the difference, the smaller the number of periods that it takes to complete the process during which agents bring the holdings of the high-deficit currency down to zero.

4.6. Recurrent Currency Crisis

In the dynamics of currency crisis in the emerging markets, there is usually no apparent reason for a sudden shift in investors' expectations.[14] Models of speculative attacks where a currency crisis can take place due to the existence of sunspot equilibria [e.g., Cole and Kehoe (1996), Jeanne and Masson (2000)] do not show how investors coordinate on a currency crisis path. Arifovic and Masson (1999) describe an evolutionary model that results in recurrent episodes of currency crisis that are driven solely by changes in investors' beliefs. The economy consists of a population of n risk-neutral investors who make portfolio decisions, that is, how much of their wealth W to invest in an emerging market at the rate r_t at time period t. Investors invest the rest of their wealth in the U.S. market at the constant rate of return, r^*. At each t, investor i, $i \in \{1, \ldots, n\}$, is characterized by a probability of devaluation π_t^i, $\pi_t^i \in [0, \pi_{\max}]$. That represents the expectation of how likely devaluation in the following period is. Let $\bar{\pi}_t$ be the average of these expectations. The expected amount of devaluation, δ^e, is equal across investors and constant over time.

Since this is a model with heterogeneous beliefs and no explicit assumption of optimizing behavior, the no-arbitrage condition cannot determine the value of r_t. Instead, it is assumed that emerging market banks set the interest rate on bank deposits. They use the average of investors' expectations as a measure of the expected value of devaluation. Thus, the interest rate on emerging market deposits r_t is set equal to the U.S. rate plus a weighted average of the expected rate of devaluation:

$$r_t = (1 + r^*)(1 + \rho) \prod_{i=1}^{n} \left(1 + \pi_t^i \delta^e\right)^{\frac{1}{n}} - 1, \quad \textbf{(29)}$$

where ρ is a constant risk premium on the return on emerging market deposits. Investor i's rule for making a portfolio decision is given by $\lambda_t^i = 0$ or 1 as $(1+r^*) \gtrless$

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

$(1 + r_t)/[(1 + \rho)(1 + \pi_t^i \delta^e)]$. Let $\overline{\lambda}_t$ be the average portfolio decision. At time t, the amount of emerging market deposits held by all foreign investors is $D_t = \sum_{i=1}^{n} \lambda_t^i W$. Note that in a representative-agent model of rational, risk-neutral investors, the portfolio fraction value is indeterminate.

The underlying model is a balance-of-payments model with the following equations describing the change in reserves $R_t = R_{t-1} + T_t + D_t - (1 + r_{t-1})D_{t-1}$ and the trade balance $T_t = \alpha + \beta T_{t-1} + \varepsilon_t$. where ε is assumed to be normally distributed with mean zero and variance σ^2.

Provided that R_t is above 0, there is no devaluation at t; that is, $\delta_t = 0$. However, if reserves would otherwise be negative, there is a devaluation that reduces the amount repaid on borrowing undertaken at $t - 1$ so that reserves at t equal zero.

Individual rates of return that investors earn determine *fitness* at time t. Thus, a fitness value of expectation i, μ_t^i, is equal to $\mu_t^i = r_t - \delta_t$ if investor i invested her wealth in the emerging market, and to $\mu_t^i = r^*$ if she invested in the U.S. market. A population of rules is updated every period using the two previously described operators: imitation and experimentation. Experimentation is performed with probability p_{ex} by drawing a random number from the uniform distribution, in the interval $[0, \pi_{\max}]$.

Simulations exhibit recurrent devaluations: extended periods of $\delta_t = 0$ are followed by instances of devaluation, $\delta_t > 0$, which take place over several periods. During the periods when $\delta_t = 0$, $\overline{\pi}_t$ and, consequently, r_t are decreasing while $\overline{\lambda}_t$ and R_t are increasing. On the other hand, during the periods of $\delta_t > 0$, $\overline{\pi}_t$, and r_t are increasing, R_t is equal to zero, and $\overline{\lambda}_t$ is decreasing. Devaluations are triggered by reversal in the general pattern of falling values of $\overline{\pi}_t$.

Note that, in any given period, as long as $\pi_t^i < \overline{\pi}_t$, investor i will invest in the emerging market and as long as $\delta_t = 0$ will earn the return $r_t > r^*$. Thus, any individual investor has to be more optimistic than the average investor in order to invest in the emerging market. In the absence of devaluation, the evolutionary dynamics drive the value of $\overline{\pi}_t$ down and increase the proportion of investors in the emerging market. However, eventually, $\overline{\pi}_t$ becomes small enough and enters into the region where a reversal of its decline occurs. This reversal occurs because there is a limit to the number of investors, so that increasing optimism cannot continue indefinitely. Once $\overline{\pi}_t$ starts increasing, investors become more and more pessimistic about emerging markets. To invest in an asset with the rate of return r^*, they have to be more pessimistic than the average investor. Eventually, as $\overline{\pi}_t$ becomes high enough, another reversal occurs and it starts decreasing again. Thus, the observed dynamics look very much like the dynamics of currency crisis observed in actual markets where there is no apparent reason for a sudden shift in investors' expectations and a withdrawal of deposits from the emerging market.

In addition to simulations, Arifovic and Masson consider two simplified versions of the model. The first model is identical to the one used in simulations except that the number of investors is infinite. The analysis shows that, if $\overline{\pi}_t$ becomes too low, and gets into the range of values $[0, \pi_{\max}/2]$, this results in a reversal with

$\bar{\pi}_{t+1} - \bar{\pi}_t > 0$. This triggers a crisis with several periods of devaluation that result in the increasing values of $\bar{\pi}_t$. However, as $\bar{\pi}_t$ enters into the region where its values are higher than $\pi_{\max}/(1 + p_{\text{ex}})$, this triggers another reversal in its behavior and the crisis is halted.

The second is a model with infinite number of investors and only two types of expectations, π^l and π^h, corresponding to a low and high expected probability of devaluation. Those with π_h invest in the safe asset, and those with π_l invest in the emerging market. Arifovic and Masson characterize the behavior of the change in $\bar{\lambda}_{t+1}$ over time and show that, as it is increasing, it reaches a critical, maximum value of λ^* after which there is no further increase in its value, and (in the absence of large trade surpluses), reserves decline over time and lead to devaluation. After devaluation, the value $\bar{\lambda}_{t+1}$ drops to the value of p_{ex} and then starts increasing again. Thus, this model results in a cyclical behavior of $\bar{\lambda}_{t+1}$ that is independent of the initial conditions.

5. CONCLUDING REMARKS

Two questions frequently arise regarding the implementation of evolutionary algorithms, genetic algorithms in particular, in economic modeling. The first is the issue of interpretation of learning at the level of population and the second is the question of interpretation and motivation for the use of the crossover operator. For example, Fudenberg and Levine (1998) note that applications of GAs have tended to assume that an entire population of players jointly implements a GA, rather than each individual player implementing a GA, and that this implementation poses problems of how to motivate and interpret this type of learning, particularly with respect to the information required for application of crossover. They suggest that, instead, individual agents could implement GAs.

These models have been primarily used as models of *social* learning where an entire population(s) evolves through imitation, exchange of ideas, and experimentation. However, as some applications show, these algorithms also can be used as models of *individual learning*, where evolution takes place on a set of competing beliefs of an indivdiual agents. (The examples are the use of the multiple-population GA in the cobweb model and the use of classifier systems in the search model of money and in the artificial stock market.) Which evolutionary paradigm is more appropriate depends on context and on a particular model. At the macroeconomic level, it seems plausible that, over time, learning takes place at the level of the economy, that is, that agents observe each other's decisions and imitate those agents that have been successful in the past. Notions of imitation of successful firms or investors have been around in economic literature for a long time. Social learning represents explicit modeling of these notions.[15] On the other hand, if the objective of research is the examination of strategic interactions in game-theoretic framework, then individual learning might be a more appropriate paradigm.

The question of motivation and interpretation of crossover can be answered in two ways. One is that it is used to capture the idea of exchange of information

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

between agents, and the second is that the evolutionary process can be represented with imitation (reproduction) and experimentation (mutation) only. The question remains what the impact of crossover is. Are there are going to be qualitative differences in the results of simulations with and without crossover?

Evolutionary algorithms have proven successful in addressing a number of issues in macroeconomic environments. They are useful devices in the selection of equilibria. When evidence from the experiments with human subjects is used for evaluation, they perform well in capturing the main features of the experimental data (e.g., cobweb model, single currency and two-currency overlapping generations economy, search model of money).

Related to the issue of equilibrium selection is the learnability of equilibria. In other words, how hard is it to learn certain equilibria? Interesting results that come out of the study of evolutionary algorithms show that there are equilibria that are easier to learn and others that cannot be learned. One of the examples is the search model of money, where agents always coordinate on the fundamental equilibrium, or what seems to be close to the fundamental equilibrium allocations, and are not able to learn how to coordinate on the speculative equilibria. The other is the overlapping generations model with periodic and chaotic equilibrium trajectories, where genetic algorithm coordinates on equilibria of low order, and never selects high-order equilibria which again appear to be hard to learn. Both of these results are supported by evidence from the experiments with human subjects.

In addition to equilibrium selection, the use of evolutionary modeling has also been useful in providing a description of the transitional dynamics in a model of growth with human capital accumulation. Moreover, this type of model can account for different timing of development takeoffs in case of the economies with identical initial conditions, and for persistence in differences in per capita income across the economies. The results on transitional dyanamics in a growth model suggest another important area of research that has not been explored so far. It is the behavior of these models in the environments where there are sudden policy changes. Transitional, evolutionary dynamics could be quite different from the RE predictions, particularly in the economies with multiple equilibria.

Competition of different forecasting rules is another issue that has been examined. This competition in a coevolutionary environment brings more stability to the economy (cobweb model) and can evolve populations of heterogenous forecasts that result in identical, equilibrium decision rules, which suggests that agents need not have identical forecasts to coordinate on an equilibrium path (n-period overlapping generations economy).

These models also have had success in capturing features of the actual time series of asset prices that a number of asset-pricing models and models of exchange rate have failed to account for (e.g., models of mutual fund investors, artificial stock market exchange rate, and currency crisis). The analysis showed that time series generated by the artificial stock market overall match the actual time series much better than RE versions of the asset-pricing models. In general, RE models cannot capture the features of the actual asset-price time series and cannot explain the observed volume of trading, its persistence, and its co-movement with volatility of asset prices.

The observed volatility of the exchange rate in a two-currency model matches the features observed in the experiments with human subjects. The persistence of volatility is the feature of the actual time series of exchange rates that other models of the exchange-rate behavior have failed to account for. An evolutionary model of currency crisis generates dynamics that look very much like the dynamics of the actual currency crisis.

Another interesting finding from the long-run learning dynamics different from rational expectations equilibrium outcomes is that the frequency of learning can qualitatively change the dynamics of the system. More frequent updating leads to persistent volatility of flows in and out of mutual funds, and to complex stock market dynamics. Similarly, higher intensity of choice in the cobweb model results in chaotic trajectories.

One of the challenges that the research in this area faces is the extension of evolutionary models to the general equilibrium type of economies with multiple markets. The main issue is the one of determination of prices. These models cannot take advantage of computing prices through simultaneous determination of agents' optimal decisions and market-clearing conditions. Instead, the calculation of prices has to be explicitly modeled by describing a bargaining process or some other equilibrating mechanism. This adds an extra layer of complexity on top of the dynamics that tend to be quite complicated anyway. However, this obstacle will have to be overcome if these models are to be more widely used in the general equilibrium setting.

As already noted, agents' beliefs and actions affect the price levels, which in turn affect agents' payoffs in models with evolutionary learning. These self-referential systems with heterogenous beliefs that evolve over time result in complicated dynamics that cannot be characterized analytically. However, techniques similar to those used in evolutionary game theory could be used in some of the models to examine the local stability of equilibria under the evolutionary dynamics. In addition, the analysis of the asymptotic behavior of the simplified versions of models can improve our understanding of simulations and provide support for the observed behavior (e.g., a cobweb model with two types of predictors, model of currency crisis). Finally, the competition of different rules, where their survival depends on performance, is an important feature that distinguishes evolutionary learning from statistical learning. This feature can be used to gain further insights into the evolutionary dynamics by examining the ways in which algorithms interact with economic environments in which they are implemented.

NOTES

1. There is a large number of applications of these algorithms in other areas of economics and in game theory. For an overview, see Dawid (1999).
2. For an introduction to GAs see Goldberg (1989) or Michalewicz (1996).

https://doi.org/10.1017/S1365100500016059 Published online by Cambridge University Press

3. Chen and Yeh (1994) applied *genetic programming* to the cobweb model. They found that the evolutionary process modeled using the genetic programming gets in the neighborhood of the REE for both the stable and the unstable case.

4. For further details on how the condition is derived, see Dawid (1999).

5. Note that the least-squares estimate of the inflation rate was obtained by regression of prices on past prices. Evans and Honkapohja (1999) note that the failure of least-squares learning to converge might be the result of this particular learning scheme. Since the price level in either steady state is a trended series, whereas the inflation rate is not, it would be more natural to estimate the inflation rate by its sample mean.

6. The *E*-stability analysis [Evans and Honkapohja (1995a)] selects the two-cycle over the steady state in this model. This result is interesting in light of the result obtained with the application of the genetic algorithm: The *E*-stability selection criterion correctly predicts the outcome of the genetic algorithm adaptation in this model.

7. The stability of steady state and periodic equilibria under learning dynamics has been the subject of a number of studies, e.g., Guesnerie and Woodford (1991) and Evans and Honkapohja (1995a,b).

8. See Van Huyck et al. (1994).

9. Goldberg (1989) and Sargent (1993) provide good descriptions of classifier systems.

10. The selection also can be made randomly with probabilities proportional to classifiers' strengths.

11. More generally, a performance measure that is a weighted average of past realized net profits can be used instead of the last-period profits. The choice of the length of the horizon and the weights is going to affect the dynamics of the system.

12. Brock and Hommes (1998b) use the same framework to study the behavior of investors using different predictors in a simple mean-variance model and show the possibility for the emergence of chaotic trajectories.

13. Notice that there is no operator that corresponds to crossover. It has been omitted to allow a greater degree of analytical tractability.

14. For example, Eichengreen et al. (1996) failed to identify significantly worse macroeconomic fundamentals in periods of crisis versus periods of noncrisis.

15. Note that the framework of evolutionary game theory is also based on the specification of learning dynamics at the level of population.

REFERENCES

Allen, F. & R. Karjalainen (1999) Using genetic algorithms to find technical trading rules. *Journal of Financial Economics* 5, 245–271.

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.

Arifovic, J. (1998) Stability of equilibria under genetic-algorithm adaptation: An analysis. *Macroeconomic Dynamics* 2, 1–22.

Arifovic, J. (1999) Performance of Rational and Boundedly Rational Agents in a Model with Persistent Fluctuations of the Exchange Rate. Manuscript, Simon Fraser University.

Arifovic, J. (in press) Evolutionary dynamics of currency substitution. *Journal of Economic Dynamics and Control*.

Arifovic, J. & P. Masson (1999) Heterogeneity and Evolution of Expectations in a Model of Currency Crisis, manuscript.

Arifovic, J., J. Bullard & J. Duffy (1997) The transition from stagnation to growth: An adaptive learning approach. *Journal of Economic Growth* 2, 185–209.

Arthur, B., J. Holland, B. LeBaron, R. Palmer & P. Tayler (1997) Asset pricing under endogenous expectations in an artificial stock market. In B. Arthur, S. Durlauf, and D. Lane (eds.), *The Economy as an Evolving Complex System II*, pp. 15–43. SFI Studies in the Sciences of Complexity, Vol. 27. Reading, MA: Addison-Wesley.

Azariadis, C. & A. Drazen (1990) Threshold externalities in economic development. *Quarterly Journal of Economics* 105, 501–526.

Basci, E. (1999) Learning by imitation. *Journal of Economic Dynamics and Control* 23, 1569–1585.

Bernasconi, M. & O. Kirchkamp (1999) Why Monetery Policy Matters—An Experimental Study of Saving, Inflation and Monetary Policies in an Overlapping Generations Model. Working Paper, available at http://wwww.sfb504.uni-mannheim.de/ kirchkamp/noi.html.

Brock, W., J. Lakonishok & B. LeBaron (1992) Simple technical trading rules and the stochastic properties of stock returns. *Journal of Finance* 47, 1731–1764.

Brock, W. & C. Hommes (1998a) A rational route to randomness. *Econometrica* 65, 1059–1097.

Brock, W. & C. Hommes (1998b) Heterogenous beliefs and routes to chaos in a simple asset pricing model. *Journal of Economic Dynamics and Control 22*, 1235–1274.

Bullard, J. (1994) Learning equilibria. *Journal of Economic Theory* 64(2): 468–485.

Bullard, J. & J. Duffy (1998a) A model of learning and emulation with artificial adaptive agents. *Journal of Economic Dynamics and Control 22*, 179–207.

Bullard, J. & J. Duffy (1998b) Learning and the stability of cycles. *Macroeconomic Dynamics* 2, 22–48.

Bullard, J. & J. Duffy (1999) Using genetic algorithms to model the evolution of heterogenous beliefs. *Computational Economics* 13, 41–60.

Campbell, J.Y. & R. Shiller (1988) Stock prices, earnings, and expected dividends. *Journal of Finance* 63, 661–676.

Chen, S.-H. & C.-H. Yeh (1994) In P. Angeline & K.E. Kinnear, Jr. (eds.), *Genetic Programming, Learning, and the Cobweb Model*. Ch. 22. Advances in Genetic Programming 2. Cambridge, MA: MIT Press.

Cole, H. & T. Kehoe (1996) A self-fulfilling model of Mexico's 1994–95 debt crisis. *Journal of International Economics* 41, 309–330.

Dawid, H. (1999) *Adaptive learning by genetic algorithms, analytical results and applications to economic models*, 2nd rev. ext. ed. Berlin: Springer-Verlag.

Dawid, H. & M. Kopel (1998) The appropriate design of a genetic algorithm in economic applications exemplified by a model of the cobweb type. *Journal of Evolutionary Economics* 8, 297–315.

Duffy, J. & J. Ochs (1999) Emergence of money as a medium of exchange: An experimental study. *American Economic Review* 89, 847–877.

Eichengreen, B., A. Rose & C. Wyplosz (1996) Contagious Currency Crises. NBER working paper 5681.

Evans, G. & S. Honkapohja (1995a) Adaptive learning and expectational stability: An introduction. In A. Kirman and M. Salmon (eds.), *Learning and Rationality in Economics*, pp. 102–126. Oxford: Basil Blackwell.

Evans, G. & S. Honkapohja (1995b) Local convergence of recursive learning to steady states and cycles in stochastic nonlinear models. *Econometrica* 63:195–206.

Evans, G. & S. Honkapohja (1999) Learning dynamics. In J. Taylor & M. Woodford (eds.), *Handbook of Macroeconomics*, vol. 1, pp. 449–542. Amsterdam, New York: North-Holland: Elsevier.

Franke, R. (1998) Coevolution and stable adjustment in the cobweb model. *Journal of Evolutionary Economics* 8, 383–406.

Frankel, G. & A. Rose (1995) Empirical research on nominal exchange rates. In G. Grossman & K. Rogoff (eds.), *Handbook of International economics*, vol. 3, pp. 1689–1729. Amsterdam: North-Holland.

Fudenberg, D. & D. Levine (1998) *Theory of Learning in Games*. Cambridge, MA: MIT Press.

Goldberg, D. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. New York: Addison-Wesley.

Guesnerie, R. & M. Woodford (1991) Stability of cycles with adaptive learning rules. In W.A. Barnett, B. Cornet, C.D'Aspremont, J. Gabszewicz & A. Mas-Collel (eds.), *Equilibrium Theory and*

Applications, Proceedings of the Sixth Inernational Symposium in Economic Theory and Econometrics, pp. 247–296. Cambridge, UK: Cambridge University Press.

Grandmont, J.-M. (1985) On endogenous competitive business cycles. *Econometrica* 53, 995–1045.

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

Jeanne, O. & P. Masson (2000) Currency crisis, sunspots, and Markov-switching regimes. *Journal of International Economics* 50, 327–350.

Kareken, J. & N. Wallace (1981) On the indeterminacy of equilibrium exchange rates. *Quarterly Journal of Economics* 96, 207–222.

Kiyotaki, N. & R. Wright (1989) On money as a medium of exchange. *Journal of Political Economy* 97, 924–954.

Koza, J.R. (1992) *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.

LeBaron, B., W. Arthur & R. Palmer (1999) Time series properties of an aritifical stock market. *Journal of Economic Dynamics and Control* 23, 1487–1516.

Lettau, M. (1997) Explaining the facts with adaptive agents: The case of mutual fund flows. *Journal of Economic Dynamics and Control* 21, 1117–1147.

Lettau, M. & H. Uhlig (1999) Rules of thumb versus dynamic programming. *American Economic Review* 89, 148–174.

Lim, S.S., E.C. Prescott & S. Sunder (1994) Stationary solution to the overlapping generations model of fiat money: Experimental evidence. *Empirical Economics* 19, 255–277.

Lucas, R.E., Jr. (1986) Adaptive behavior and economic theory. *Journal of Business* 59, 401–426.

Maddison, A. (1982) *Phases of Capitalist Development*. New York: Oxford University Press.

Manski, C. & D. McFadden (1981) *Structural Analysis of Discrete Data with Econometric Applications*. Cambridge, MA: MIT Press.

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

Marimon, R., E. McGrattan & T.J. Sargent (1989) Money as a medium of exchange in an economy with artificially intelligent agents. *Journal of Economic Dynamics and Control* 14, 329–373.

Marimon, R. & Sunder, S. (1993) Indeterminacy of equilibria in a hyperinflationary worlds: Experimental evidence. *Econometrica* 61, 1073–1107.

Meese, R. & K. Rogoff (1983) Empirical exchange rate models of the seventies. *Journal of International Economics* 14, 3–24.

Michalewicz, Z. (1996) *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed. New York: Springer-Verlag.

Muth, J. (1961) Rational expectations and the theory of price movements. *Econometrica* 29, 315–335.

Neely, C.J., P. Weller & R. Dittmar (1997) Is technical analysis in the foreign exchange market profitable? A genetic programming approach. *Journal of Financial and Quantitative Analysis* 32, 405–426.

Parente, S.L. & E.C. Prescott (1993) Changes in the wealth of nations. *Quarterly Review*, Federal Reserve Bank of Minneapolis (Spring), 3–16.

Sargent, T. (1993) *Bounded Rationality in Macroeconomics*. Oxford: Clarendon Press.

Staudinger, S. (1998) Money as a Medium of Exchange—An Analysis with Genetic Algorithms. Working paper, Technical University, Vienna.

Van Huyck, J.B., J.P. Cook & R.C. Battalio (1994) Selection dynamics, asymptotic stability, and adaptive behavior. *Journal of Political Economy* 102, 975–1005.

Wellford, C.P. (1989) A Laboratory Analysis of Price Dynamics and Expectations in the Cobweb Model. Discussion paper 89-15. University of Arizona, Tuscon.