*Macroeconomic Dynamics*, **2**, 1998, 1–21. Printed in the United States of America.

**ARTICLES**

# STABILITY OF EQUILIBRIA UNDER GENETIC-ALGORITHM ADAPTATION: AN ANALYSIS

**JASMINA ARIFOVIC**
*Simon Fraser University*

The stability of stationary equilibria under genetic-algorithm adaptation is studied in an overlapping-generations economy with a unique, monetary steady-state and two-period cyclic equilibria. Agents use the genetic algorithm to update their labor-supply decisions. The results of simulations and the analysis of the dynamics show that the two-period cycle is stable and that the steady-state equilibrium is unstable under genetic-algorithm adaptation. The dynamics are driven primarily by the wealth and substitution effects that lead to the algorithm's convergence to the two-period cycle.

**Keywords:** Stability, Genetic Algorithms

## 1. INTRODUCTION

The results of the analysis of stability of equilibria under learning dynamics can be used as an equilibrium selection criterion in economic environments with multiplicity of equilibria. The stability of steady-state and periodic equilibria under learning dynamics has been the subject of a number of studies, e.g., Grandmont (1985), Grandmont and Laroque (1986), Woodford (1990), Guesnerie and Woodford (1991), Evans and Honkapohja (1995a,b). 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 $k$-period cycle, agents have to use $k$-order adaptive learning rules.

This paper studies the stability of steady-state and periodic equilibria in an overlapping-generations economy with fiat money in which agents use the genetic algorithm[1] to update their labor-supply decisions. The model has a unique monetary steady-state and two-period cyclical equilibria.

When used in economic modeling, the genetic algorithm describes the evolution of a population of rules, representing different possible beliefs, in response to experience. In a parallel to population genetics, these rules undergo a selection process whereby more successful ones become more numerous in the population.

I thank John Duffy and George Evans for helpful comments and suggestions. Address correspondence to: Jasmina Arifovic, Department of Economics, Simon Fraser University, Burnaby, British Columbia V5A 1S6, Canada; e-mail: arifovic@sfu.ca.

© 1998 Cambridge University Press 1365-1005/98 $9.50 

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

The rules are subjected to random mutations and to recombination of their parts. These changes contribute to the diversity of the population by introducing new rules (new ideas) to be tried.

The behavior of the genetic algorithm has been studied in a number of macroeconomic environments [Marimon et al. (1989); Bullard and Duffy (1994, 1996); Arifovic (1995, 1996); Arifovic et al. (1997)]. Bullard and Duffy (1996) and Dawid (1996) conduct simulations in the environments in which periodic equilibria exist. There are several advantages in modeling of agents' adaptation in this way. Genetic algorithms impose low requirement on the computational ability of economic agents. They allow for modeling heterogeneity of agents' beliefs. Survival of decision rules depends on their performance, measured by the payoff that agents receive by employing them. Finally, genetic algorithms perform better than models with rational agents or alternative models of adaptive behavior in explaining the features observed in experimental economies [see, e.g., Andreoni and Miller (1995); Arifovic (1995, 1996)]. The main disadvantage of the genetic-algorithm studies, including the ones performed within the class of models with periodic equilibria, has been that the results are based entirely on simulations.[2]

This paper examines the global stability[3] of the monetary steady state and two-period cyclical equilibria. The analysis, which uses notions of stability similar to those of the evolutionary game theory, shows that the two-period cyclical equilibrium is globally stable and that the steady-state equilibrium is unstable under genetic-algorithm adaptation. Regardless of the initial conditions, the genetic algorithm converges to the two-period cycle. The initial conditions affect only the speed of convergence to the two-period cycle.

The paper is organized as follows. The description of the model of the stationary-equilibrium outcomes is given in Section 2. That section also presents the expectational-stability conditions for the model's stationary equilibria. Application of the genetic algorithm is described in Section 3. The results of simulations are given in Section 4, and the analysis of the dynamics is given in Section 5. Concluding remarks are given in Section 6.

## 2. DESCRIPTION OF MODEL

The economy is inhabited by agents who live for two periods. They work when young and consume when old. The production function is linear, and one unit of labor yields one unit of output. Fiat money is the only means for saving. The stock of nominal money, $H_t$, is constant over time ($H_t = H$ for all $t$).

The maximization problem of an agent born at time $t$ is given by

$$\max ac_{t+1} - \tfrac{1}{2}bc_{t+1}^2 - n_t,$$
$$m_t \leq p_t n_t,$$
$$p_{t+1}c_{t+1} \leq m_t,$$

where $a > 0, b > 0$, $n_t$ is labor supply when young, $n_t \in [0, 1]$, $c_{t+1}$ is consumption when old, $p_t$ is the nominal price level at time $t$, $m_t$ are the nominal money

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

balances accumulated at the end of time period $t$, and $p_{t+1}$ is the nominal price level at $t+1$.

Given the above functional form,[4] the first-order condition is

$$a\frac{p_t}{p_{t+1}} - b\left(\frac{p_t}{p_{t+1}}\right)^2 n_t - 1 = 0, \tag{1}$$

and the optimal labor supply, $\hat{n}_t$, is given by

$$\hat{n}_t = \frac{aR_t - 1}{bR_t^2}, \tag{2}$$

where $R_t$ is the gross rate of return on money holdings between periods $t$ and $t+1$, i.e., $R_t = p_t/p_{t+1}$. This implies the optimal second-period consumption

$$\hat{c}_{t+1} = \frac{a - 1/R_t}{b}. \tag{3}$$

In equilibrium, $y_t = n_t = H/p_t$. It follows that, in equilibrium,

$$R_t = n_{t+1}/n_t. \tag{4}$$

Then, the equilibrium law of motion for the labor supply is

$$n_t = \frac{a(n_{t+1}/n_t) - 1}{b(n_{t+1}/n_t)^2}. \tag{5}$$

Given the chosen parameterization ($a \in [3.2, 3.4]$, $b \in [3.2, 3.4]$, $a = b$), there exist a unique monetary steady-state equilibrium and a two-period cyclical equilibrium. In the steady-state equilibrium, $n_t = n_{t+1} = n^*$, $p_t = p_{t+1} = p^*$, and $R_{t-1} = R_t = R^* = 1$, for all $t$. From (4) and (5), the steady-state value of labor supply then is equal to

$$n^* = (a-1)/b. \tag{6}$$

Let $n^{*,1}$ and $n^{*,2}$ denote the cyclical equilibrium labor-supply decisions. The two-period cycle corresponds to $n^{*,1} = \phi(n^{*,2})$ and $n^{*,2} = \phi(n^{*,1})$ and is obtained as the solution to the following set of equations:

$$\begin{aligned} n^{*,1} &= n^{*,2}(a - bn^{*,2}), \\ n^{*,2} &= n^{*,1}(a - bn^{*,1}). \end{aligned} \tag{7}$$

The cyclical equilibrium nominal prices are denoted by $p^{*,1}$ and $p^{*,2}$ and the rate of return on money holdings by $R^{*,1}$ and $R^{*,2}$, respectively.

The local stability of the stationary equilibria, above, under learning dynamics can be determined by calculating the expectational-stability ($E$-stability) conditions.[5] Evans and Honkapohja (1995a) give the $E$-stability conditions for the nonstochastic overlapping-generations economy with fiat money with a unique monetary steady state and two-period cycles. The optimal behavior of agents

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

depends on the forecast price $p^e_{t+1}$. In a temporary equilibrium, agents observe $p_t$ and $n_t$ so that using $H = p_{t+1}n_{t+1}$, they can equivalently try to forecast $n^e_{t+1}$. For a given $n^e_{t+1}$, temporary equilibrium then is characterized by $n_t = F(n^e_{t+1})$ or

$$n_t = \frac{a\left(n^e_{t+1}/n_t\right) - 1}{b\left(n^e_{t+1}/n_t\right)^2}. \tag{8}$$

Suppose that the perceived law of motion is a steady state, i.e., is of the form

$$n_t = n^e_{t+1}.$$

Then, the actual law of motion is

$$n_t = F\left(n^e_{t+1}\right) = n^e_{t+1}\left(a - bn^e_{t+1}\right). \tag{9}$$

The condition for the weak $E$-stability of the steady state is that $F'(n^*) < 1$. Because $F'(n^*) < 1$, this condition is satisfied. If agents assume that they are in the steady state and estimate the steady-state parameter as the average of the past observed values, the learning algorithm will converge to the steady state. However, the steady state is not strongly $E$-stable, i.e., it is not robust to overparameterization. The condition for strong $E$-stability is that $|F'(n^*)| < 1$. Because $|F'(n^*)| > 1$, this condition is not satisfied. This implies that, if there exists a two-period cycle and if agents' perceived law of motion is a two-period cycle, the economy will not converge to the steady state from some initial points near $p^*$ and the learning algorithm may instead converge to the two-period cycle.

The condition for the weak $E$-stability of the two-period cycle is that $F'(n^{*,2})$ $F'(n^{*,1}) < 1$. Because this condition is satisfied, the two-period cycle is weakly $E$-stable. The learning algorithm that estimates the odd-period value of $n^e_{t+1}$ as the average of past labor-supply values in odd periods, and the even-period value of $n^e_{t+1}$ as the average of past labor-supply values in even periods, converges to the two-period cycle. The two-period cycle of this model is also robust to overparameterization, i.e., it is strongly $E$-stable. The rational two-cycle is strongly $E$-stable if it is stable when regarded as a $2n$-cycle for any $n$. The condition for strong $E$-stability is that $|F'(n^{*,2})F'(n^{*,1})| < 1$. 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 genetic-algorithm adaptation in this model. It also raises the question of whether the link between $E$-stability and stability under genetic-algorithm learning holds more generally.

## 3. DESCRIPTION OF GENETIC ALGORITHM

At each integer point in time $t \geq 1$, there are two populations of binary strings: one being the new population of generation $t$, the young; the other being the population of generation $t - 1$, the old. A population of generation $t$ consists of $N$ binary strings that represent decision rules of $N$ agents of generation $t$. In each time

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

period $t$, only the *young* strings, members of generation $t$, take action that affects the market outcomes. At the same time, the market outcomes of these actions affect the payoffs received by the *old*, the strings of generation $t-1$.

A binary string $i, i \in [1, N]$, of length $l$ encodes a labor-supply decision that an agent makes when young. Decoding of a binary string is performed in the following way:

$$x_{i,t} = \sum_{k=1}^{l} a_{i,t}^{k} 2^{k-1}, \tag{10}$$

where $a_{i,t}^{k}$ is the value (0, 1) taken at the $k$th position in the $i$th string. The integer $x_{i,t}$ is normalized to obtain a real value of the labor supply $n_{i,t}$, $n_{i,t} \in [0, 1]$. Thus,

$$n_{i,t} = x_{i,t}/\bar{K}, \tag{11}$$

where $\bar{K} = 2^{30} - 1$. The range of values of labor supply is between 0 (if all bits of a binary string are equal to zero) and 1 (if all bits of a binary string are equal to 1).

The aggregate savings at $t$, $S_t$, are equal to the aggregate supply of output and are determined by adding individual amounts of output produced:

$$S_t = \sum_{i=1}^{N} n_{i,t}. \tag{12}$$

The nominal price level $p_t$ then is given by

$$p_t = H/S_t. \tag{13}$$

Using the nominal price level, the money balances that agent $i$ carries over to period $t+1$ are computed as

$$m_{i,t} = p_t n_{i,t}. \tag{14}$$

At time period $t+1$, the consumption by agent $i$ of generation $t$ is determined by

$$c_{i,t+1} = m_{i,t}/p_{t+1}. \tag{15}$$

At the end of time period $t+1$, the utilities of members of generation $t$ are calculated. They depend on individual labor-supply decisions and on the nominal prices at $t$ and $t+1$. These utilities, in turn, determine the fitness values of the respective decision rules. The fitness of a decision rule represented by a binary string $i$ of generation $t$, $\mu_{i,t}$, is given by the value of agent $i$'s utility at the end of period $t+1$:[6]

$$\mu_{i,t} = ac_{i,t+1} - \tfrac{1}{2} bc_{i,t+1}^{2} - n_{i,t}. \tag{16}$$

Decision rules are updated using four operators: reproduction, crossover, mutation, and election.

*Reproduction* makes copies of individual strings. It promotes well-performing strings over time. The strings that have high fitness values relative to those of their

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

peers obtain more copies over time and take larger fractions of the population. Reproduction is implemented through $N$ *tournaments*. Each tournament is performed in the following way: A pair of binary strings is selected randomly with replacement. The fitness values of two strings are compared and the one with the higher fitness value is chosen. In case of a tie, a string is chosen randomly. The chosen string contributes a copy to the *mating* pool. These steps are repeated $N$ times, and $N$ copies of strings enter into a mating pool to undergo application of other genetic operators.

*Crossover* operates on members of the mating pool. First, a pair of strings is selected randomly. Second, an integer number $k$ is selected from $[1, l-1]$, again at random. Two new strings are formed by swapping the set of values to the right of the position $k$. An example that illustrates how crossover is implemented is given below. Suppose that the two decision rules that are randomly paired are given by the following binary strings:

$$\begin{cases} 000011011011110101001101000111 10 \\ 111111100000101000111100011100 00 \end{cases}.$$

Suppose that $k = 10$ is chosen randomly from the interval $[1, l-1]$, where $l = 30$. Next, two binary strings are divided into two parts with the dividing point to the right of bit position 10:

$$\begin{cases} 0000110110 \mid 1110101001101000111 10 \\ 1111111000 \mid 0101000111100011100 00 \end{cases}.$$

The two parts to the right of position 10 then are swapped, and the resulting two strings are

$$\begin{cases} 000011011010101000111100011100 00 \\ 111111100001110101001101000111 10 \end{cases}.$$

The total number of pairs that are selected in $N/2$ (where $N$ is an even integer). Crossover takes place on each pair with the probability $p_{\text{cross}}$.

*Mutation* randomly alters the value of a position within a string to the other value taken by the binary alphabet: $0 \to 1$, $1 \to 0$. The probability of mutation, $p_{\text{mut}}$, is independent and identical across the positions.

The implementation of the mutation operator is combined with the implementation of the crossover operator in the following way: Each pair of strings selected from the mating pool for crossover is recorded as a pair of parents of two new offspring. If crossover occurs, the strings are recombined. The mutation operator is applied to bits of the two strings that resulted from the crossover exchange. Once the application of the mutation operator is completed, two strings are recorded as two new offspring. If crossover does not occur, the bits of the two parent strings are exposed to mutation. The resulting strings are recorded as two new offspring of a given pair of parents.

*Election* [Arifovic (1994)] tests newly generated offspring before they can enter into a population of generation $t+2$. A binary string $i$, $i \in [1 \cdots N]$, that belongs

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

to a new offspring $i$ is decoded into a real number that represents offspring $i$'s labor-supply decision, $n_{i,t}$. Based on this decision and on the rate of return between periods $t$ and $t+1$, a second-period consumption value for offspring $i$ is computed. Labor supply and second-period consumption then are used to compute offspring $i$'s *potential* fitness value, $\mu^{p}_{i,t+3}$.

Election is applied in the following way. Two offspring and two parents (recorded at the crossover and mutation stages of the genetic-algorithm application) are ranked, from the highest to the lowest, on the basis of their fitness values. Potential fitness values of the offspring and actual fitness values of the parents then are compared. The two strings that have the first and the second rank are chosen and placed into the population of a new generation. In case of a tie between a parent and an offspring, an offspring is placed into the population.

A population of decision rules of agents of generation $t$ is generated in the following way: First, the application of the reproduction operator to the members of generation $t-2$ yields a population of $N$ copies. Then, the crossover and mutation operators are applied to generate new decision rules to be tried. Finally, new offspring decision rules are subjected to the election operator test. Offspring decision rules that pass the test, together with parents that are more fit than their offspring, form a new population of decision rules of agents of generation $t$. The values of $n_{i,t}$ for $i \in [1, N]$ are computed. The savings of generation $t$ then are used to compute the nominal price level at $t$. This price and individual labor supply determine individual holdings of money balances that are carried over to period $t+1$. The nominal price $p_t$ and individual money holdings of the members of generation $t-1$ then are used to compute individual second-period consumption values, $c_{i,t}$'s, of the members of generation $t-1$. At the end, the utilities of members of generation $t-1$ are determined. The population of generation $t+1$ is generated through the application of the genetic-algorithm operators to the members of generation $t-1$.

## 4. SIMULATIONS AND ANALYSIS OF DYNAMICS

To examine the behavior of the genetic algorithm in this environment, three types of simulations were conducted: randomly initialized simulations, simulations initialized in the neighborhood of the steady-state equilibrium, and simulations initialized in the neighborhood of the two-period cycle. Five different sets of parameter values $a$ and $b$ were used for each type of initialization. These values are given in Table 1, which also contains the values of labor supply, nominal prices, and rates of return in the stationary equilibria for these parameter values. Per-capita money supply was equal to 10 in all simulations.

The parameters of the genetic algorithm were set using the recommended values found in the artificial-intelligence literature. The population size $N$ was set equal to 30, which usually is taken as the minimum number of strings required for effective search. The length of a binary string, $l$, which determines the fineness of the search grid, was set to 30 bits. The probability of crossover, $p_{\text{cross}}$, was set equal to 0.6, and the probability of mutation, $p_{\text{mut}}$, was set equal to 0.033.

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

**TABLE 1.** Parameter values and stationary equilibrium values of labor supply, nominal price levels, and rates of return

| Set | $a$ | $b$ | $n^*$ | $p^*$ | $R^*$ | $n^{*,1}$ | $p^{*,1}$ | $R^{*,1}$ | $n^{*,2}$ | $p^{*,2}$ | $R^{*,2}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3.2 | 3.2 | 0.69 | 14.54 | 1 | 0.80 | 12.51 | 0.64 | 0.51 | 19.49 | 1.56 |
| 2 | 3.25 | 3.25 | 0.69 | 14.44 | 1 | 0.812 | 12.31 | 0.61 | 0.50 | 20.19 | 1.64 |
| 3 | 3.3 | 3.3 | 0.70 | 14.35 | 1 | 0.82 | 12.14 | 0.58 | 0.48 | 20.86 | 1.72 |
| 4 | 3.35 | 3.35 | 0.70 | 14.25 | 1 | 0.83 | 14.25 | 0.56 | 0.46 | 21.5 | 1.79 |
| 5 | 3.4 | 3.4 | 0.71 | 14.17 | 1 | 0.84 | 11.87 | 0.54 | 0.45 | 22.12 | 1.86 |

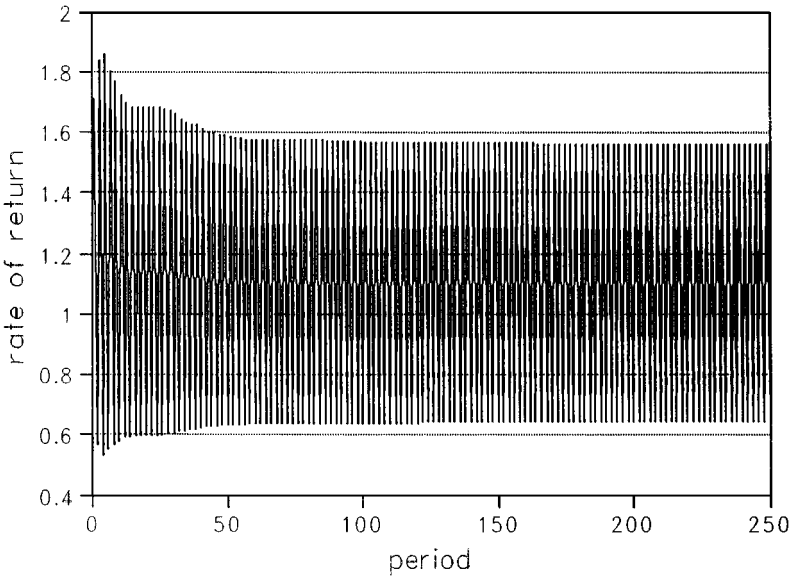



**FIGURE 1.** Rate of return in randomly initialized simulation: $a = b = 3.2$.

These values are those that are recommended by Back (1996) and Michalewicz (1996). Every combination of the overlapping-generations parameter values and the genetic operator values was simulated using multiple runs with different seed values for the random-number generator to ensure the robustness of the results to different pseudo-random-number sequences.

A total of 75 simulations was conducted (3 types of initialization $\times$ 5 sets of parameter values $\times$ 5 random-number seed values). All simulations converged to the two-period cycle. The convergence implies identical decision rules within each of the two populations. The rules of one population all take the value $n^{*,1}$, and the rules of the other population all take the value $n^{*,2}$. The rate of return in a simulation that was randomly initialized with the first set of parameter values is presented in Figure 1. After relatively large initial fluctuations, the rate of return settles into the pattern of the two-period cycle around generation 200.[7]

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

**TABLE 2.** Evaluation and updating of two genetic-algorithm populations

| | Population (pop) | | | |
|---|---|---|---|---|
| $t =$ | Young | Old | Updating | Evaluation |
| 1 | 1 | 2 | — | — |
| 2 | 2 | 1 | — | 1 |
| 3 | 1 | 2 | 1 | 2 |
| 4 | 2 | 1 | 2 | 1 |

In describing the events that take place during the genetic-algorithm adaptation, one of the two genetic-algorithm populations is denoted as pop 1, the average labor supply of its members by $\bar{n}_t^1$, the rate of return relevant for evaluating the fitness values of its decision rules by $R_t^1$, and the labor supply that is optimal (given $R_t^1$) by $\hat{n}_t^1$. Rules that are members of this population are *evaluated* at the end of every *even* period and they are *updated* at the beginning of every *odd* period. The other population is denoted as pop 2, the average labor supply of its members by $\bar{n}_t^2$, the rate of return relevant for evaluating the fitness of its rules by $R_t^2$, and the labor supply that is optimal (given $R_t^2$) by $\hat{n}_t^2$. Members of pop 2 are *evaluated* at the end of every *odd* period and *updated* at the beginning of every *even* period. The pattern of evaluation and updating for the two populations in the first six periods of the simulation is given in Table 2.

Thus, $R_t^j$ denotes the rate of return on money holdings between $t$ and $t + 1$, where superscript $j$, $j \in [1, 2]$, denotes the population that receives this rate of return on the money holdings of its members; $\bar{n}_t^j$, $j \in [1, 2]$, denotes the average labor supply of young agents of pop $j$ at time $t$; and $\hat{n}_t^j$, $j \in [1, 2]$, denotes the optimal labor supply at $t$, given the rate of return $R_t^j$. (Note that this is the ex post optimal labor supply that is determined once $R_t^j$ is calculated.)

The analysis of the genetic-algorithm adaptation begins with a consideration of the main elements that affect the evolutionary dynamics. It continues with a detailed description of the events that take place in two simulations, one of which was initialized in the neighborhood of the steady-state equilibrium and the other one that was initialized in the neighborhood of the two-period cycle. The examination of the observed patterns of behavior suggests that the dynamics are defined by the impact of the wealth and intertemporal substitution effects on the evolution of decision rules. The analysis presented in the following section concludes that the steady-state equilibrium is unstable under genetic-algorithm adaptation, whereas the two-period cycle is globally stable.

The main elements that define the dynamics of the evolutionary adaptation are the difference between the average actual labor supply and the optimal labor supply, the impact of the wealth effect, and the impact of the substitution effect.

A fitness value of a rule $n_{k,t}^j \neq \hat{n}_t^j$, $j = 1, 2, k \in [1, N]$, which is evaluated at the end of $t + 1$ at $R_t^j$, will be lower than the fitness value associated with

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

$\hat{n}_t^j$. *The difference between the average actual labor supply,* $\bar{n}_t^j$, *and the optimal labor supply,* $\hat{n}_t^j$, $|\bar{n}_t^j - \hat{n}_t^j|$, determines the upper bound on the amount of average change that may take place among the population decision rules.

The larger the difference, the larger the change that takes place. The election operator accepts an average newly generated offspring rule that takes the value between the population-average labor supply and the labor supply that would have been optimal given the rate of return. If $\hat{n}_t^j < \bar{n}_t^j$, $\bar{n}_{t+2}^j \leq \bar{n}_t^j$. The election operator accepts an average new offspring such that

$$\hat{n}_t^j \leq \bar{n}_{t+2}^{j,o} \leq \bar{n}_t^{j,p},$$

where $\bar{n}_t^{j,p}$ is an average parent rule of generation $t$, and $\bar{n}_{t+2}^{j,o}$ is an average offspring rule that will be accepted as a member of generation $t+2$. On the other hand, if $\hat{n}_t^j > \bar{n}_t^j$, then $\bar{n}_{t+2}^j \geq \bar{n}_t^j$. The election operator accepts an average new offspring such that

$$\bar{n}_t^{j,p} \leq \bar{n}_{t+2}^{j,o} \leq \hat{n}_t^j.$$

Given that $R_t^j = \bar{n}_{t+1}^i / \bar{n}_t^j$, $i, j = 1, 2$, $i \neq j$, changes in both $\bar{n}_t^i$ and $\bar{n}_t^j$ affect the level of $R_t^j$. Increases in $\bar{n}_t^j$ have negative effect on $R_t^j$, and increases in $\bar{n}_t^i$ have positive effect. Thus, if $\bar{n}_t^i$ is increasing and $\bar{n}_t^j$ is decreasing, $R_t^j$ is increasing and $R_t^i$ is decreasing and vice versa. If both $\bar{n}_t^i$ and $\bar{n}_t^j$ are increasing or decreasing, the direction in which $R_t^j$ moves depends on the magnitude of relative changes in $\bar{n}_t^i$ and $\bar{n}_t^j$. For example, if there are increases (or decreases) in both $\bar{n}_{t+1}^i$ and $\bar{n}_t^j$, then

$$R_t^j \uparrow \quad \text{if} \quad \Delta\bar{n}_{t+1}^i > \Delta\bar{n}_t^j,$$
$$R_t^j \downarrow \quad \text{if} \quad \Delta\bar{n}_{t+1}^i < \Delta\bar{n}_t^j,$$

where $\Delta\bar{n}_{t+1}^i = (\bar{n}_{t+3}^i - \bar{n}_t^i)/\bar{n}_{t+1}^i$ and $\Delta\bar{n}_t^j = (\bar{n}_{t+2}^j - \bar{n}_t^j)/\bar{n}_t^j$.

In turn, changes in $R_t^j$ affect $\hat{n}_t^j$ and, consequently, $\bar{n}_t^j$ as well. The way in which $\hat{n}_t^j$ is affected depends on whether the wealth or the substitution effect dominates. Let $R^b$ be the value of the rate of return that separates the regions of dominance of the wealth effect and of the substitution effect. For $R_t^j \geq R^b$, the wealth effect dominates the substitution effect, and a decrease in $R_t^j$ results in an increase in the optimal labor supply $\hat{n}_t^j$. On the other hand, for $R_t^j < R^b$, the substitution effect dominates the wealth effect and a decrease in $R_t^j$ results in a decrease in $\hat{n}_t^j$. Increases in $\hat{n}_t^j$ are followed by increases in $\bar{n}_{t+2}^j$ if $\bar{n}_t^j < \hat{n}_t^j$. Decreases in $\hat{n}_t^j$ are followed by decreases in $\bar{n}_{t+2}^j$ if $\bar{n}_t^j > \hat{n}_t^j$. To illustrate the difference in the impact of the wealth and substitution effect, Table 3 shows what happens to the optimal labor supplies, average labor supplies, and rates of return of both pop 1 and pop 2 as a result of a decrease in $R_t^1$.

Two simulations that are described below were conducted using the first set of the parameter values, $a = b = 3.2$. Given the utility function and the parameter

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

**TABLE 3.** Effects of changes in rate of return

*A. Decrease in $R_t^1$ in region where wealth effect dominates*:

$$\left(\hat{n}_t^1 > \bar{n}_t^1, \hat{n}_t^2 < \bar{n}_t^2\right)$$

$$\downarrow R_t^1 \rightarrow \uparrow \hat{n}_t^1 \rightarrow \uparrow \bar{n}_{t+2}^1 \rightarrow \downarrow R_{t+2}^1$$

$$\searrow \uparrow R_{t+1}^2 \rightarrow \downarrow \hat{n}_{t+1}^2 \rightarrow \downarrow \bar{n}_{t+3}^2 \rightarrow \downarrow R_{t+2}^1$$

*B. Decrease in $R_t^1$ in region where substitution effect dominates*:

$$\left(\hat{n}_t^1 < \bar{n}_t^1, \hat{n}_t^2 > \bar{n}_t^2\right)$$

$$\downarrow R_t^1 \rightarrow \downarrow \hat{n}_t^1 \rightarrow \downarrow \bar{n}_{t+2}^1 z \rightarrow \uparrow R_{t+2}^1$$

$$\searrow \downarrow R_{t+1}^2 \rightarrow \uparrow \hat{n}_{t+1}^2 \rightarrow \uparrow \bar{n}_{t+3}^2 \rightarrow \uparrow R_{t+2}^1$$

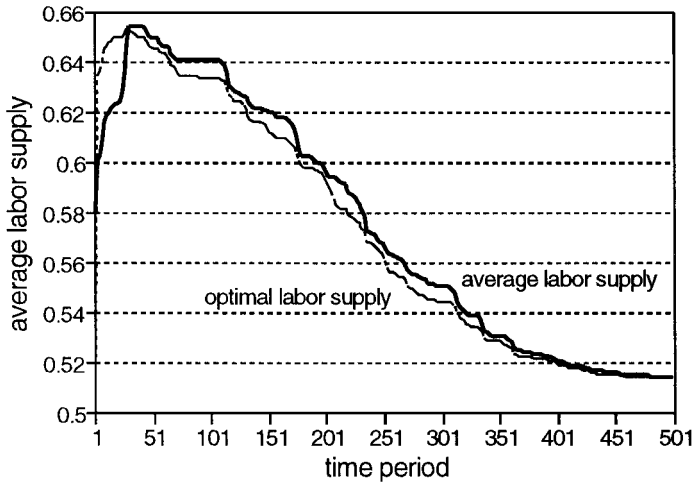



**FIGURE 2.** Average and optimal labor supply of pop 1 in simulation initialized in the neighborhood of the steady state: $a = b = 3.2$.

values used in these simulations, the wealth effect dominates the substitution effect for $R_t^j \geq R^b = 0.625$, $j = 1, 2$. Both stationary equilibria are in the region where the wealth effect dominates the substitution effect.

### 4.1. Simulation 1

The first simulation was initialized in the neighborhood of the steady-state equilibrium and it lasted for 500 periods. The behavior of labor supply and rates of return is given in Figures 2–5: $\bar{n}_t^1$ in Figure 2, $R_t^1$ in Figure 3, $\bar{n}_t^2$ in Figure 4, and $R_t^2$ in Figure 5. Because $R_t^1$ and $R_t^2$ never fall below $R^b$ during the simulation, the wealth effect dominates the substitution effect for the entire duration of the simulation. At the beginning of simulation, two populations of binary strings are generated. Every member of pop 1 decodes to the value of 0.6 and every member of pop 2 decodes to the steady-state value of 0.6875.

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

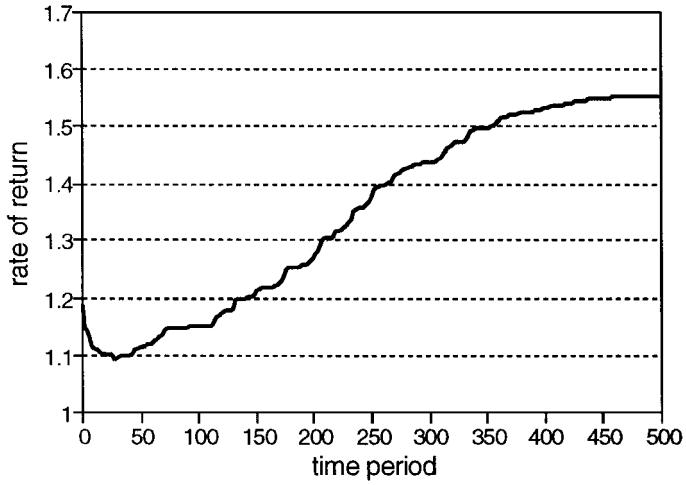



**FIGURE 3.** Rate of return of pop 1 in simulation initialized in the neighborhood of the steady state: $a = b = 3.2$.

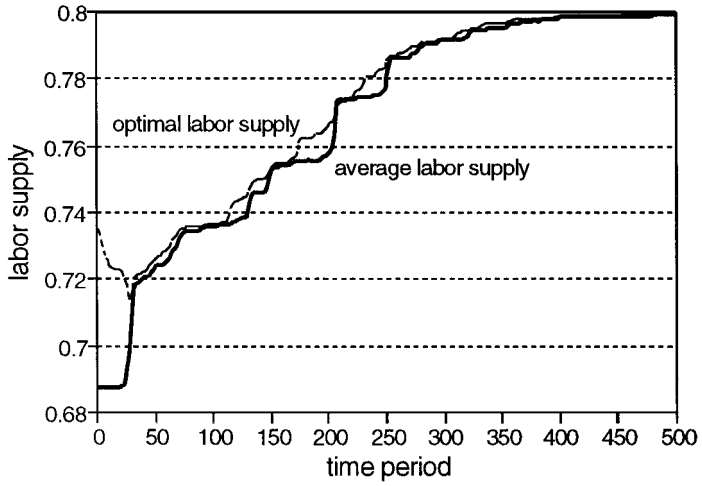



**FIGURE 4.** Average and optimal labor supply of pop 2 in simulation initialized in the neighborhood of the steady state: $a = b = 3.2$.

The rules of pop 1 evolve from the value of 0.6 to one of the values of the two-period cycle, $n^{*,2} = 0.513$. The following events take place during their adjustment: Between $t = 3$ and $t = 28$, $\hat{n}_t^1 > \bar{n}_t^1$. This results in the increases in $\bar{n}_t^1$ during this period. These increases in $\bar{n}_t^1$ have negative effect on $R_t^1$ which, as a result, is decreasing during this period. Note that the level of $R_t^1$ also is affected by changes in $\bar{n}_t^2$. Between $t = 2$ and $t = 17$, $\bar{n}_t^2$ is not changing and thus has no

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

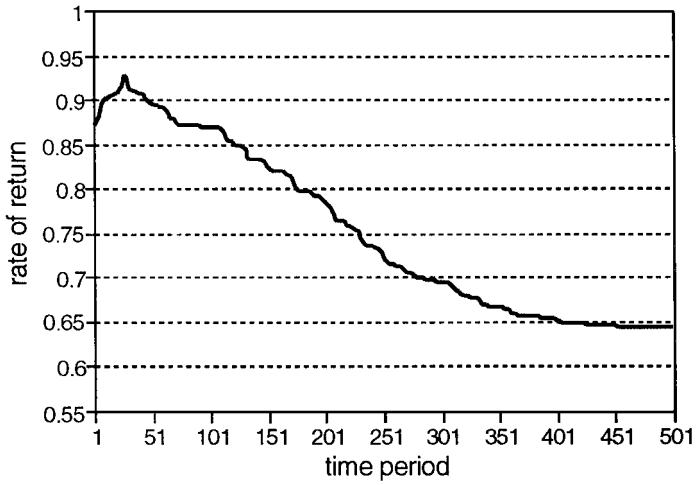



**FIGURE 5.** Rate of return of pop 2 in simulation initialized in the neighborhood of the steady state: $a = b = 3.2$.

effect on $R_t^1$. After $t = 17$, $\bar{n}_t^2$ starts increasing. However, the positive effects of these increases on $R_t^1$ are outweighed by the negative effects of increases in $\bar{n}_t^1$.

At $t = 28$, the positive effect of $\bar{n}_t^2$ outweighs the negative effect of $\bar{n}_t^1$ and $R_t^1$ increases. In response to this change, $\hat{n}_t^1$ drops down and, after $t = 28$, becomes lower than $\bar{n}_t^1$. As a result, $\bar{n}_t^1$ starts decreasing. Further, decreases in $\bar{n}_t^1$ have positive impact on $R_t^1$, which starts increasing. (During this period, $\bar{n}_t^2$ is increasing and these increases also contribute to the increases in $R_t^1$.) Rising $R_t^1$ in turn lowers $\hat{n}_t^1$ and $\bar{n}_t^1$ further. This process is uninterrupted until the convergence to the values of the two-period cycle, $\bar{n}_t^1 = n^{*,2}$ and $R_t^1 = R^{*,2}$, which occurs at $t = 499$.

At the same time, rules of pop 2 are evolving toward the other of the two sets of values of the two-period cyclical equilibrium, $n^{*,1} = 0.7995$. Between $t = 3$ and $t = 17$, $\hat{n}_t^2 > \bar{n}_t^2$. During this period, no change in $\bar{n}_t^2$ occurs. Changes in $R_t^2$ take place as a result of the changes in $\bar{n}_t^1$. Because $\bar{n}_t^1$ is increasing during this period, $R_t^2$ also is increasing. In addition, $\hat{n}_t^2$ is decreasing in response to increases in $R_t^2$. At $t = 18$, $\bar{n}_t^2$ starts increasing toward $\hat{n}_t^2$, which remains higher than $\bar{n}_t^2$. Increases in $\bar{n}_t^2$ continue until $t = 27$. Between $t = 18$ and $t = 27$, $\bar{n}_t^2$ and $R_t^2$ are increasing. This happens because $\bar{n}_t^1$ also is increasing during this period. Increases in $\bar{n}_t^1$ have positive impact on $R_t^2$ and outweigh the negative impact of the increases in $\bar{n}_t^2$ during this period.

After $t = 27$, as $\bar{n}_t^1$ starts decreasing, $R_t^2$ decreases as well. In response, $\hat{n}_t^2$ increases. Because $\hat{n}_t^2 > \bar{n}_t^2$, $\bar{n}_t^2$ increases. In turn, $R_t^2$ decreases further. This process continues until the end of simulation. During this period, $\bar{n}_t^1$ is increasing and thus it also has negative effect on $R_t^2$. Finally, $R_t^2$ reaches the value of $R^{*,1}$ and $\bar{n}_t^2$ the value of $n^{*,1}$ at $t = 500$.

After the short initial stage during which $\bar{n}_t^1$ is increasing and $\bar{n}_t^2$ is not changing, the second stage of adjustment begins in which the wealth effect determines the

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

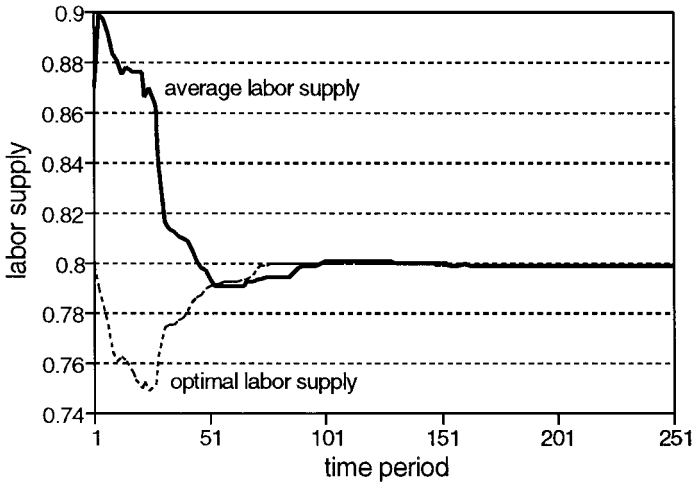



**FIGURE 6.** Average and optimal labor supply of pop 1 in simulation initialized in the neighborhood of the two-period cycle: $a = b = 3.2$.

dynamics of the simulation. During this stage, $\hat{n}_t^1$ and $\overline{n}_t^1$ are decreasing, and $R_t^1$ is increasing. At the same time, $\hat{n}_t^2$ and $\overline{n}_t^2$ are increasing, and $R_t^2$ is decreasing. The two populations of rules move away from the steady-state equilibrium over time and converge to the values of the two-period cycle.

It is worth pointing out that the dynamics do not change if a simulation is initialized closer to the equilibrium. For example, a distance from the monetary steady-state equilibrium will be much shorter if only one rule of pop 1 differs from $n^*$ while the rest of $N - 1$ members of pop 1 decode to $n^*$. (Members of pop 2 all decode to $n^*$.) A simulation initialized in this way would take longer to move away from the steady-state equilibrium, but there would be no qualitative difference in the behavior of the genetic algorithm.

### 4.2. Simulation 2

The second simulation was initialized in the neighborhood of the two-period cycle. It lasted for 250 periods. The behavior of labor supplies and rates of return is given in Figures 6–9: $\overline{n}_t^1$ in Figure 6, $R_t^1$ in Figure 7, $\overline{n}_t^2$ in Figure 8, and $R_t^2$ in Figure 9. At the beginning of simulation, rules of pop 1 decode to the value of 0.9. Rules of pop 2 decode to the lower of the 2 two-period-cycle values of labor supply, $n_2^2 = n^{*,2} = 0.513$.

At the end of $t = 2$, $R_1^1 = 0.56$. Because $R_1^1$ is below $R^b = 0.625$, the rules of pop 1 are in the region of dominance of the substitution effect. Between $t = 3$ and $t = 25$, $\overline{n}_t^1$ is decreasing because $\hat{n}_t^1 < \overline{n}_t^1$. Decreases in $\overline{n}_t^1$ have positive effect on $R_t^1$, but they are outweighed by the negative effects of the decreases in $\overline{n}_t^2$ that take place during this time. Thus, $R_t^1$ also is decreasing. Because the rules are in the

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

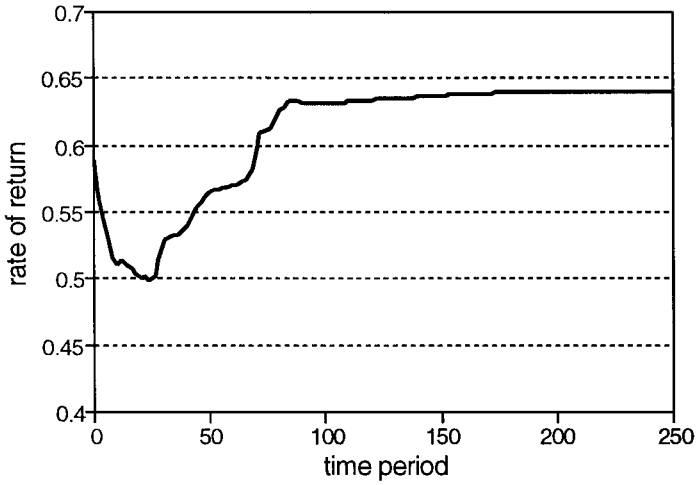



**FIGURE 7.** Rate of return of pop 1 in simulation initialized in the neighborhood of the two-period cycle: $a = b = 3.2$.

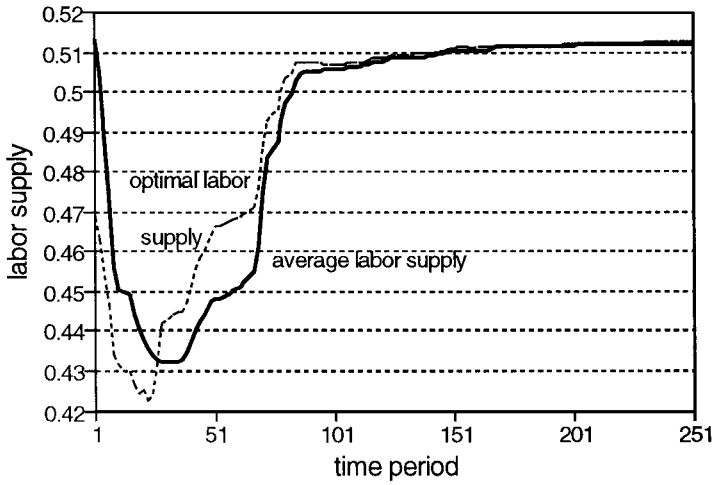



**FIGURE 8.** Average and optimal labor supply of pop 2 in simulation initialized in the neighborhood of the two-period cycle: $a = b = 3.2$.

region where the substitution effect dominates, decreases in $R_t^1$ result in decreases in $\hat{n}_t^1$. Decreases in $\hat{n}_t^1$ cause further decreases in $\bar{n}_t^1$.

At $t = 26$, the effect of $\bar{n}_t^1$ on $R_t^1$ outweighs the effect of $\bar{n}_t^2$ and $R_t^1$ increases for the first time. Because this is still the area of dominance of the substitution effect, $\hat{n}_t^1$ increases in response to this increase in $R_t^1$. However, because $\bar{n}_t^1$ is larger than $\hat{n}_t^1$, $\bar{n}_t^1$ continues to decrease. This, in turn, has a positive impact on $R_t^1$ and a negative impact on $R_t^2$.

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

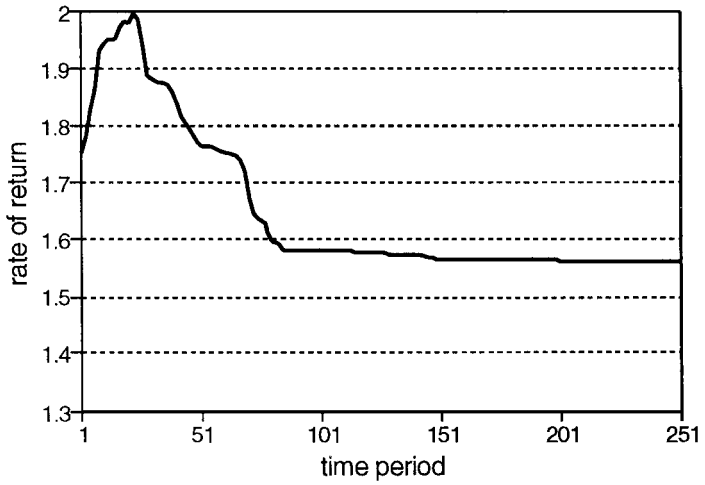



**FIGURE 9.** Rate of return of pop 2 in simulation initialized at the two-period cycle: $a = b = 3.2$.

At $t = 108$, $R_t^1$ takes the value greater than $R^b$ and pop 1 enters the region of dominance of the wealth effect. At this point, both $\bar{n}_t^1$ and $\hat{n}_t^1$ are greater than $n^{*,1}$. The rate of return, $R_t^1$, continues to increase. Now that the wealth effect dominates, these changes result in decreases in $\hat{n}_t^1$ and subsequent decreases in $\bar{n}_t^1$. Increases in $\bar{n}_t^2$ and decreases in $\bar{n}_t^1$ during this period both have positive effects on $R_t^1$. Increases in $R_t^1$ trigger further decreases in $\hat{n}_t^1$ and $\bar{n}_t^1$. This process continues until $\bar{n}_t^1$ and $R_t^1$ reach the values of the two-period cycle, $n^{*,1}$ and $R^{*,1}$, at $t = 249$.

In the meantime, the following dynamics take place in pop 2, which is in the region of dominance of the wealth effect during the entire simulation. Between $t = 3$ and $t = 28$, $\hat{n}_t^2 < \bar{n}_t^2$. This results in successive decreases in $\bar{n}_t^2$. Decreases in $\bar{n}_t^2$ bring about increases in $R_t^2$ which lead to further decreases in $\hat{n}_t^2$ and $\bar{n}_t^2$. Decreases in $\bar{n}_t^1$ that take place at the same time have negative effect on $R_t^2$, but this effect is outweighed by the positive effect of the decreases in $\bar{n}_t^2$.

At $t = 29$, the negative effect of $\bar{n}_t^1$ exceeds the positive effect of $\bar{n}_t^2$ and it lowers $R_t^2$. This also results in $\bar{n}_t^2 < \hat{n}_t^2$. From this point on, $R_t^2$ is decreasing, and $\hat{n}_t^2$ and $\bar{n}_t^2$ are increasing steadily until the end of the simulation. (During the periods of the increases in $\bar{n}_t^1$, their positive impact on $R_t^2$ is outweighed by a negative impact of decreases in $\bar{n}_t^2$.) The decision rules of pop 2 converge to $n^{*,2}$ at $t = 250$.

In this simulation, both the wealth and the substitution effect exert influence on the dynamics of the simulation. Whereas pop 2, which starts at $n^{*,2}$, stays in the area of dominance of the wealth effect, pop 1, which starts with labor-supply values greater than $n^{*,1}$, passes through regions of both the wealth- and the substitution-effect dominance. It starts in the region of the wealth-effect dominance, which drives the average labor supply further up and its rate of return further down. This continues until the population enters the region of dominance of the substitution effect when the process of further increases in the average labor supply and decreases

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

in the rate of return is halted. The dynamics bring the population back to the region of dominance of the wealth effect. The population now experiences decreases in the average labor supply and increases in its rate of return as it approaches $n^{*,1}$ from below.

## 5. CONVERGENCE AND STABILITY RESULTS

*If the genetic algorithm converges, it converges to a stationary equilibrium point.* Levels of labor supply different from the stationary equilibrium values result in rates of return that imply levels of optimal labor supply different from the actual ones. As long as there is a difference between actual and optimal levels of labor supply, the average labor supply will continue to change. For example, suppose that all members of pop 1 decode to 0.4, and all the members of pop 2 decode to 0.9. The populations will not remain at these points for the following reason: These values of labor supply result in $R_t^1 = 2.25$ and $R_t^2 = 0.44$, but, at these rates of return, $\hat{n}_t^1 = 0.38$ and $\hat{n}_t^2 = 0.66$. Because $\bar{n}_t^1 \neq \hat{n}_t^1$ and $\bar{n}_t^2 \neq \hat{n}_t^2$, the election operator will accept new offspring with the level of labor supply closer to the optimal level. Acceptance of new, different rules will result in changes in the rates of return, which induce further changes in the values of the optimal and average labor supplies. The only points for which $n_{k,t}^j = \bar{n}_t^j = \hat{n}_t^j$ for all $k \in [1, N]$ are the stationary equilibrium points, $n^s$, the steady state, and the two-period cycle. If there is no $n_{k,t}^j \neq n^s$ such that $\mu_t(n_{k,t}^j, R^s) = \mu(n^s, R^s)$, i.e., the fitness value of $n_{k,t}^j$ equals the fitness value of $n^s$, no new offspring that decodes to the values different from $n^s$ will be accepted into the population. As a result, there will be no further changes in the populations of decision rules and they will remain at the stationary equilibrium points.

Suppose that the genetic algorithm is in the two-period cycle in a simulation with $a = b = 3.2$. Rules of pop 1 decode to $n^{*,1} = 0.7995$. Rules of pop 2 decode to $n^{*,2} = 0.5130$. These values of labor supply result in the rates of return $R_t^1 = R^{*,1} = 0.642$ and $R_t^2 = R^{*,2} = 1.558$. In turn, the optimal labor supplies for these two rates of return are $\hat{n}_t^1 = 0.7995$ and $\hat{n}_t^2 = 0.5130$. Because these optimal values are equal to the actual levels of labor supply, no new offspring that decode to values different from the stationary equilibrium ones will be accepted.

Thus the algorithm will not converge to the values that are not stationary equilibrium values. The *stability* of stationary equilibria under genetic-algorithm adaptation will determine a stationary equilibrium that is *selected*. The genetic algorithm always converges to the two-period cycle because it is globally stable under the genetic-algorithm adaptation. It diverges away from the steady-state equilibrium because it is unstable under genetic-algorithm adaptation.

The *stability of a stationary equilibrium* under genetic-algorithm adaptation can be examined in a way similar to the analysis of the equilibrium stability used in the evolutionary game theory. Such analysis consists of examining what happens when there is an $\epsilon$ departure from the stationary-equilibrium value. A rule that deviates from the stationary value by the amount $\epsilon$ is inserted into a population of

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

rules that decode to the stationary-equilibrium values. The stationary equilibrium is *stable* under genetic-algorithm adaptation if the algorithm converges back to the equilibrium after initial adjustment. On the other hand, the stationary equilibrium is *unstable* if the insertion of a deviating rule results in the divergence of the genetic algorithm away from the stationary equilibrium.

The *steady-state equilibrium* is *unstable* under genetic-algorithm adaptation. Suppose that at $t = T$, the decision rules of both populations decode to the steady-state values of labor supply. At $t > T$, a rule $n^1_{k,t}$ is inserted into pop 1, such that $|n^1_{k,t} - n^*| = \epsilon > 0$. Suppose that $n^1_{k,t} > n^*$. Then, $R^1_t < R^*$, and $\hat{n}^1_t$ increases. In response to an increase in $\hat{n}^1_t$, $\overline{n}^1_t$ increases. At the same time, $R^2_t > R^*$ and $\hat{n}^2_t$ decreases. In response to a decrease in $\hat{n}^2_t$, $\overline{n}^2_t$ decreases. This results in further decrease in $R^1_t$ and increase in $R^2_t$. Because both $R^1_t$ and $R^2_t$ are in the region where the wealth-effect dominates, the process that is initiated in response to this difference in the rates of return takes the two populations away from the steady-state equilibrium. If the initial populations start further away from the steady-state equilibrium, the wealth effect will prevent them from getting into the neighborhood of the steady-state equilibrium and converging to it.

The *two-period cycle* is *stable* under genetic-algorithm adaptation. Suppose, that at $t = T$, the decision rules of pop 1 and pop 2 decode to the values of labor supply in the two-period cycle. In particular, members of pop 1 decode to $n^{*,1}$ and members of pop 2 decode to $n^{*,2}$, with rates of return $R^{*,1}$ and $R^{*,2}$. Suppose that, at time $t$, $t > T$, a rule $n^1_{k,t}$ is inserted into pop 1, such that $|n^1_{k,t} - n^{*,1}| = \epsilon > 0$. The remaining $N - 1$ rules decode to $n^{*,1}$. Suppose that $n_{k,t} > n^{*,1}$ and that $R^{*,1}$ is in the region where the wealth effect dominates. The adjustment of the genetic algorithm that takes place is described in steps 1–4.

1. At the start, the deviating rule will have a negative effect on $R^1_t$ and a positive effect on $R^2_t$. Decreases in $R^1_t$ are accompanied by increases in $\hat{n}^1_t$ and $\overline{n}^1_t$. At the same time, increases in $R^2_t$ are accompanied by decreases in $\hat{n}^2_t$ and $\overline{n}^2_t$. Once $R^1_t$ reaches $R^b$, pop 1 enters into the region of dominance of the substitution effect, and $\hat{n}^1_t$ and $\overline{n}^1_t$ start decreasing in response to further decreases in $R^1_t$.
2. Because both $\overline{n}^1_t$ and $\overline{n}^2_t$ are decreasing, changes in $\overline{n}^2_t$ have to be larger than changes in $\overline{n}^1_t$, $R^2_t$, and $\overline{n}^2_t$ in order to sustain an increasing sequence of $R^2_t$ and a decreasing sequence of $R^1_t$. Because the rate of decrease of $\overline{n}^2_t$ is greater than the rate of decrease of $\hat{n}^2_t$, the difference between $\overline{n}^2_t$ and $\hat{n}^2_t$ gets smaller and, over time, there is less room for further changes in $\overline{n}^2_t$. At the same time, because $R^1_t$ is decreasing, the rates of decrease of $R^1_t$ and $\hat{n}^1_t$ are greater than the rate of decrease of $\overline{n}^1_t$, and the difference between $\hat{n}^1_t$ and $\overline{n}^1_t$ is increasing.
   Once a change in $\overline{n}^2_t$ is not sufficient to sustain a sequence of rising $R^2_t$'s, there will be a reversal in the movements of the rates of return which begins by an increase in $R^1_t$ and a decrease in $R^2_t$. If this increase in $R^1_t$ is large enough, $R^1_t$ will pass $R^b$, and pop 1 will be back in the region of dominance of the wealth effect. In this case, a change (increase) in $R^1_t$ will result in a reduction of $\hat{n}^1_t$ and, subsequently, of $\overline{n}^1_t$. Once again, both populations are under dominance of the wealth effect. In this case, the dynamics proceed to step 4.

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

3. If the increase is not large enough, $R_t^1$ will remain lower than $R^b$, and pop 1 will stay in the region of dominance of the substitution effect. An increase in $R_t^1$ will bring an increase in $\hat{n}_t^1$. At this point, $\overline{n}_t^1 > \hat{n}_t^1$ and $\overline{n}_t^1$ continues decreasing. As $R_t^2$ decreases, $\hat{n}_t^2$ increases and $\overline{n}_t^2$ increases in response to the difference between $\hat{n}_t^2$ and $\overline{n}_t^2$. As $R_t^1$ continues increasing, it will reach the $R^b$ level and pop 1 will be in the wealth-effect region again.[8]
4. Once $R^b$ is passed, $\hat{n}_t^1$ starts decreasing with further increases in $R_t^1$. This change in direction is accompanied by decreases in $\overline{n}_t^1$. At the same time, $R_t^2$ is decreasing and $\overline{n}_t^2$ is increasing. These dynamics, driven by the wealth effect, continue until pop 1 converges to $n^{*,1}$ and pop 2 converges to $n^{*,2}$.

There are three other possible ways of introducing deviating rules into the populations of rules that decode to the two-period-cycle values of labor supply: to insert $n_{k,t}^1 < n^{*,1}$; to insert $n_{k,t}^2 < n^{*,2}$; and to insert $n_{k,t}^2 > n^{*,2}$. The dynamics of adjustment under these three scenarios are not discussed. It suffices to point out that they would be entirely driven by the wealth effect, which would bring the populations back to the values of the two-period cycle.

Finally, two studies that apply the genetic-algorithm adaptation in models with cyclical equilibria are worth commenting on. Dawid (1996) conducts genetic-algorithm simulations in a pure-endowment overlapping-generations model with constant money supply in which two-period cycles exist. Binary strings encode agents' forecast of next period's return on their savings. The simulations converged to the two-period cycle. However, Dawid's study does not analyze the stability properties of stationary equilibria under genetic-algorithm adaptation and does not examine the dynamics of evolution that result in the convergence to the two-period cycle.

Bullard and Duffy (1996) study the genetic algorithm in the overlapping-generations economies in which perfect-foresight cycles of high order exist. In their genetic-algorithm implementation, agent $i$ forecasts the next period's price by using the following rule: $p_{i,t+1}^e = p_{t-k_i-1}$ where $p_{i,t+1}^e$ is agent $i$'s forecast of the nominal price $p_{t+1}$ and $k_i$ is an integer encoded by agent $i$'s binary string. Different values of $k_i$ are consistent with perfect-foresight cycles of different order. Every value of $k_i$ is also consistent with learning steady states. The genetic-algorithm adaptation resulted in the convergence to cycles of low order. The convergence to higher-order cycles was not observed. Even though these results indicate that equilibria in this class of models may have different stability properties under genetic-algorithm adaptation than under algorithms that have been examined previously, this finding leaves open the question why the genetic algorithm exhibits this type of dynamics.

## 6. CONCLUDING REMARKS

The model analyzed in this paper illustrates the way in which the genetic algorithm can be used for the analysis of the global stability of equilibria in environments

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

characterized by the multiplicity of equilibria. The results of simulations and the subsequent analysis of the dynamics of genetic-algorithm adjustment show that the steady-state equilibrium is unstable under the genetic-algorithm adaptation and that the two-period cycle is globally stable. This implies that, regardless of the initial values, the algorithm will converge to the two-period cycle. The analysis shows that the dynamics of the genetic-algorithm adjustment are driven by the wealth and substitution effects.

In the case in which the genetic-algorithm populations start in the neighborhood of the steady-state equilibrium, the wealth effect will drive the population of decision rules away from the steady state. One of the populations steadily increases its labor supply in response to decreases in rates of return on money holdings. These increases result in further decreases of their rates of return and lead to increases in the rates of return on money holdings of the other population. As a result, the other population steadily decreases its labor supply, which in turn has a positive effect on its own rate of return and a negative effect on the rate of return of the first population. Such dynamics result in convergence to the two-period-cycle values of the labor supply.

When the populations start in the neighborhood of the two-period cycle, the wealth effect can, in the first stage, drive the population that starts in the neighborhood of the higher value of labor supply further away from the equilibrium value. In the process, this population will increase its average labor supply, thus decreasing its rate of return until it reaches the region where the substitution effect dominates. In this region, further increases in the average labor supply, in response to the decreases in the rate of return, are prevented by the substitution effect. In the second stage, the dynamics bring the population back to the region of dominance of the wealth effect where the average labor supply decreases and the rate of return increases as the population converges to the value of the two-period cycle. While the first population goes through the above changes, the other population that initially decodes to the lower value of labor supply also will move away from the equilibrium value. In the first stage, its average labor supply will decrease and its rate of return will increase. This movement is reversed in the second stage. The average labor supply will start increasing and the rate of return will start falling as the population returns to the equilibrium value.

## *NOTES*

1. The genetic algorithm was developed by Holland (1975) as a stochastic search algorithm.
2. The exceptions are Arifovic (1996) and Arifovic et al. (1997).
3. Global stability is the property of an equilibrium to which the genetic algorithm converges regardless of the initial conditions.
4. The utility function has been considered by Benhabib and Day (1982).
5. These stability conditions are calculated for the parameter values considered in the paper.
6. If a binary string decodes to the value of labor supply equal to 0, the fitness value of such a string is set to 0.
7. Note that this result does not depend on the particular way in which reproduction is implemented

https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press

in this version of the genetic algorithm. Bullard and Duffy (1994) use a *mixing* procedure to create a pool of copies of binary strings. The entire genetic stock—in this case, both genetic-algorithm populations—is used to create newborns. Bullard and Duffy note that this method substantially reduces the potential of the system to cycle near the steady state. Simulations conducted with mixing did not result in the convergence of the algorithm to the steady state. Instead, they exhibited cyclical behavior.

8. If the direction of the $R_t^1$ movement is reversed before it reaches $R^b$, and $R_t^1$ starts decreasing, the dynamics go back to step 2 and eventually get to step 3. This sequence will be repeated until the dynamics do reach step 4.

## *REFERENCES*


Andreoni, J. & J. Miller (1995) Auctions with artificial adaptive agents. *Games and Economic Behavior* 10, 39–64.

Arifovic, J. (1994) Genetic algorithm learning and the cobweb model. *Journal of Economic Dynamics and Control* 18, 3–28.

Arifovic, J. (1995) Genetic algorithms and inflationary economies. *Journal of Monetary Economics* 36, 219–243.

Arifovic, J. (1996) The behavior of the exchange rate in the genetic algorithm and experimental economies. *Journal of Political Economy* 104, 510–541.

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

Back, T. (1996) *Evolutionary Algorithms in Theory and Practice*. Oxford: Oxford University Press.

Benhabib, J. & R. Day (1982) A characterization of erratic dynamics in the overlapping generations model. *Journal of Economic Dynamics and Control* 4, 37–55.

Bullard, J. & J. Duffy (1994) Using Genetic Algorithms to Model the Evolution of Heterogenous Beliefs. Working paper, Federal Reserve Bank of St. Louis and University of Pittsburgh.

Bullard, J. & J. Duffy (1996) On Learning and the Stability of Cycles. Working paper, Federal Reserve Bank of St. Louis and University of Pittsburgh.

Dawid, H. (1996) *Adaptive Learning by Genetic Algorithms*. Berlin: Springer-Verlag.

Evans, G. & S. Honkapohja (1995a) Adaptive learning and expectational stability: An introduction. In A. Kirman & 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.

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

Grandmont, J.-M. & G. Laroque (1986) Stability of cycles and expectation. *Journal of Economic Theory* 40, 138–151.

Guesnerie, R. & M. Woodford (1991) Stability of cycles with adaptive learning rules. In W.A. Barnett, B. Cornet, C. D'Aspremont, J. Gabszewicz, and A. Mas-Colell (eds.), *Equilibrium Theory and Applications, Proceedings of the Sixth International Symposium in Economic Theory and Econometrics*, pp. 247–296. Cambridge, England: Cambridge University Press.

Holland, J. (1975) *Adaptation in Natural and Artificial Systems*: *An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Ann Arbor: University of Michigan Press.

Marimon, M., 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., S. Spear & S. Sunder (1993) Expectationally driven market volatility: An experimental study. *Journal of Economic Theory* 61, 74–103.

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

Woodford, M. (1990) Learning to believe in sunspots. *Econometrica* 58, 277–307.


https://doi.org/10.1017/S1365100598006014 Published online by Cambridge University Press