

Genetic algorithm learning and the cobweb model*

Jasmina Arifovic

McGill University, Montréal, Qué. H3A 2T7, Canada

Received March 1992, final version received December 1992

This paper presents the cobweb model in which competitive firms, in a market for a single good, use a genetic algorithm to update their decision rules about next-period production and sales. The results of simulations show that the genetic algorithm converges to the rational expectations equilibrium for a wider range of parameter values than other algorithms frequently studied within the context of the cobweb model. Price and quantity patterns generated by the genetic algorithm are also compared to the data of experimental cobweb economies. It is shown that the algorithm can capture several features of the experimental behavior of human subjects better than three other learning algorithms that are considered.

1. Introduction

Departure from the hypothesis that economic agents form rational expectations implies that a specific learning algorithm has to be employed in order to describe the way in which agents make decisions about their economic behavior. On the other hand, application of a particular algorithm faces a criticism of the arbitrariness of choice. Lucas (1986) suggests that comparison of the behavior of learning algorithms with the behavior exhibited in experimental economies with human subjects may be a possible way to address this problem.

Thus, if learning algorithms, when applied to the same economic environment, result in different behavior, observations from laboratory experiments with human subjects may be used to determine which algorithm is more successful in describing actual human behavior. In this paper, a genetic algorithm (GA), developed by Holland (1970a), is used to model learning of

Correspondence to: Jasmina Arifovic, Department of Economics, McGill University, 855 Sherbrooke Street W., Montréal, Qué. H3A 2T7, Canada.

*This paper derives from my doctoral dissertation. My special thanks go to my advisors Robert Lucas, Thomas Sargent, and Michael Woodford for their valuable help and ideas. Helpful suggestions were also received on an earlier draft from three anonymous referees.

economic agents in the cobweb model. The results obtained with the application of the GA are compared to the behavior observed in cobweb experiments with human subjects and to the results obtained in studies of other learning algorithms within the context of the same model. The objective is to examine if the GA can account for some of the results of the experimental economies which differ from the predictions of other adaptive schemes.

The cobweb model is a model of a market for a single good in which firms that are price takers make their production decision in every time period before they observe a market price. Total quantity supplied and the exogenously given demand determine the price that clears the market. The *cobweb theorem*, first formulated by Ezekiel (1938), states that (provided a firm bases its production plan on the assumption that present prices will continue and its own quantity produced will not affect the market) the market price will converge to its equilibrium value if the ratio of the demand and supply slopes is less than one (cobweb stable case). If the ratio is greater than one (cobweb unstable case), then the market price diverges away from the equilibrium.

Since then, the model has been extensively used to examine the behavior of the system under various assumptions about the way in which agents form their expectations. The rational expectations version of this model has been formulated by Muth (1961), while different versions of the cobweb model with learning have been studied, among others, by Nerlove (1958), Carlson (1969), Townsend (1978), DeCanio (1979), Frydman (1981), Brandenburger (1984), Bray and Savin (1986), Marcet and Sargent (1987), and Nyarko (1990). Holt and Williamil (1986) and Wellford (1989) simulated the model in experimental economies with human subjects. While most of these algorithms (three of which will be discussed in the second section of the paper) result in divergent behavior for the cobweb unstable case, divergent patterns were not observed in the experiments with human subjects in which cobweb economies were simulated.

The GA describes the evolution of a population of rules, representing different possible beliefs, in response to experience. The frequency with which a given rule is represented in the population indicates the degree of credence attached to it (if the population represents a single mind), or the degree to which it is accepted in a population of agents (if the population represents different agents interacting in a market). Rules whose application has been more successful are more likely to become more frequently represented in the population, through a process similar to the natural selection in population genetics. Random mutations also create new rules by changing certain features of rules previously represented in the population, thus allowing new ideas to be tried.

First application of GAs to economics was described in Miller (1986). GAs and other computer-based adaptive algorithms have been used in a variety of economic environments [for example, Miller (1989), Marimon, McGrattan,

and Sargent (1990), Rust, Palmer, and Miller (1989), Binmore and Samuelson (1990)].¹ Among others, the questions examined in these studies relate to the capability of adaptive agents to learn Nash equilibrium behavior, equilibrium selection in the environments with the multiplicity of equilibria, and the computation of equilibria in economies in which it is hard to obtain analytical solutions. The algorithms have also been useful in the examination of the behavior observed in laboratory experiments with human subjects [Crawford (1989), Miller and Andreoni (1990a), Miller and Andreoni (1990b), Arthur (1991)]. The results of these studies show that computer-based adaptive algorithms can generate behavior that corresponds to the regularities observed in experiments with human subjects which differ from the predictions of or cannot be explained by models with rational agents.

The existence of a number of results on learning and experimental behavior within the framework of the cobweb model motivated the choice of that model for GA application presented in this paper. Results of Wellford's experimental economies were used for the comparison with GA behavior. GA economic environment and GA agent's decision making were tailored in such a way as to match Wellford's experimental design. The experimental design was also a criterion for choosing three algorithms, cobweb expectations, sample average of past prices, and least squares, whose behavior is contrasted to the patterns observed in experiments and to those generated by the GA. These algorithms were studied in the cobweb environments which correspond to or can be easily mapped into environments simulated in Wellford's experimental economies.

The second objective of the paper is to examine the behavior of the cobweb model GA within two alternative designs, *single-population* and *multiple-population* design [Arifovic (1991)]. These designs are related to two possible ways in which we can think about a GA population of strings: (1) that a population of strings represents a population of agents with different opinions, each string standing for an individual agent (single-population design), or (2) that every agent has a whole population of strings, each string representing one of an agent's alternative ideas (multiple-population design).

Computer simulations of the cobweb model were conducted using two GA versions. One is the version described in Goldberg (1989) with three genetic operators: *reproduction*, *crossover*, and *mutation*, while the other, in addition to these three, includes the *election* operator [Arifovic (1991)]. The first version will be referred to as the *basic GA* and the second will be referred to as the *augmented GA*.

The paper is organized as follows. Description of the rational expectations version of the cobweb model is presented in the second section. This section also contains the results of previous work on learning and experiments within the

¹ Holland and Miller (1991) provide an overview of the applications of computer-based adaptive algorithms, including the GA, in economics.

context of the cobweb model. Section 3 describes the single-population GA application, while multiple-population GA design and results of simulations are presented in the fourth section. Section 5 is related to the comparison of the GA's behavior to Wellford's experimental results and to the behavior of cobweb expectations, sample average of past prices, and least squares learning algorithms. Concluding remarks are given in the sixth section.

2. Description of the cobweb model

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

The cost of a production of a firm i is given by

$$C_{i,t} = xq_{i,t} + \frac{1}{2}ynq_{i,t}^2, \quad (1)$$

where $C_{i,t}$ is firm i 's cost of a production for sale at time t and $q_{i,t}$ is the quantity it produces for sale at time t .

The expected profit of an individual firm, $\Pi_{i,t}^e$, is

$$\Pi_{i,t}^e = P_t^e q_{i,t} - xq_{i,t} - \frac{1}{2}yn(q_{i,t})^2, \quad (2)$$

where P_t^e is the expected price of the good at t . At $t - 1$, each firm chooses a quantity $q_{i,t}$ to maximize its expected profit $\Pi_{i,t}^e$ on the basis of its expectations, P_t^e , about the price that will prevail at time t . Thus, from the first-order conditions for profit maximization, $q_{i,t}$ is given by

$$q_{i,t} = \frac{1}{yn}(P_t^e - x). \quad (3)$$

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

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

In the rational expectations equilibrium $P_t^e = P_t$, i.e., firms' expectations about the price, P_t , of a good in period t equal the equilibrium price [Muth (1961)]. Thus,

$$x + ynq_t = A - Bnq_t,$$

where $q_t = q_{i,t}$ for all i , or

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

The objectives of the GA application are to see whether quantities produced and offered for sale by firms that are using the GA as their learning scheme will converge to this constant quantity, q^* , and how the results obtained in GA computer simulations compare to other results on learning and experimental behavior.

2.1. Application of learning algorithms

As already noted in the introduction, the cobweb model setup has been widely used in studies of learning. Three algorithms that are chosen for the comparison with the GA and experimental behavior are cobweb expectations, sample average of past prices, and least squares.

The assumption of *cobweb expectations* [Ezekiel (1938)] is that agents expect a price at t to be equal to the price at $t - 1$,

$$P_t^e = P_{t-1}.$$

A firm bases its plans for future production on the assumption that present prices will continue, and that its own production will not affect the market. The price will converge to the equilibrium price for the cobweb stable case, i.e., only if $B/y < 1$ [Nerlove (1958)]. For the cobweb unstable case ($B/y > 1$), the price sequence diverges away from the equilibrium.

If price expectation is given as a *sample average of past prices* [Carlson (1969)],

$$P_t^e = \frac{1}{t} \sum_{s=0}^{t-1} P_s,$$

the price sequence converges to the equilibrium value for both the cobweb stable case ($B/y < 1$) and the cobweb unstable case ($B/y > 1$).

In a model in which agents use *least squares* to update their price estimate, the expectation of price P_t is given by

$$P_t^e = \beta_t P_{t-1},$$

where

$$\beta_t = \frac{\sum_{s=0}^{t-1} P_s P_{s-1}}{\sum_{s=0}^{t-1} P_{s-1}^2},$$

given initial prices P_{-1} and P_0 . In each time period t , agents run regression on past values of prices to obtain an estimate of the coefficient β_t .

In the version of the cobweb model with stochastic exogenous demand and exogenous supply shock which firms observe before they make production decisions, Bray and Savin (1986) derive conditions for the convergence of the least squares estimate, β_t , to its equilibrium value (in equilibrium $\beta^* = 1$) for the cobweb stable case. For the cobweb unstable case, they conjecture that the estimate diverges away. This conjecture is supported by the results of their computer simulations of the unstable case. In all simulations they conducted, the estimate diverged away from the equilibrium.

2.2. Wellford's cobweb experiments

Wellford conducted the total of twelve experiments in which both stable and unstable cobweb cases were simulated. Each experiment had five participants and lasted for thirty periods. Sellers in the market for a single good had to make decisions on how much to offer for sale in the next period. Market clearing price was computed using exogenously given demand schedule and the total market supply, determined as the sum of individual supplies.

The results of the experiments showed that unstable cobweb treatments did not exhibit exploding patterns. Price fluctuations, within the region defined by the competitive price and Cournot price, characterized all experimental data. In some of the experiments, there were periods when the market price would reach the rational expectations equilibrium level, but did not remain there for the whole duration of any of these experiments.

Since the price paths of unstable cases exhibited greater fluctuations than those of stable ones, Wellford tested the hypothesis that the price variance across all periods was the same for both stable and unstable treatments. The hypothesis was rejected in favor of the alternative hypothesis (that the price variance of the unstable treatments exceeds that of the stable treatments).

3. Single-population genetic algorithm

A population of chromosomes, A_t , represents a collection of firms' decision rules at time period t . A firm i , $i = 1, \dots, n$, makes a decision about its production for time t using a chromosome, $A_{i,t}$ (member of population at time period t), a binary string of finite length l , written over $\{0, 1\}$ alphabet. A decoded and normalized value of a binary string i gives the value of the quantity produced by a firm i at time period t .

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

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

where $a_{i,t}^k$ is the value (0, 1) taken at the k th position in the string.

After a string is decoded, integer $x_{i,t}$ is normalized in order to obtain a real number value $q_{i,t}$, the quantity that firm i decides to produce and offer for sale at time period t :

$$q_{i,t} = x_{i,t}/\bar{K},$$

where \bar{K} is a coefficient chosen to normalize the value of $x_{i,t}$. An example of the way in which decoding and normalization work for a string of length $l = 3$ is given in appendix 1.

Fitness of a string i at time period t , $\mu_{i,t}$, is determined by the value of firms' profit earned at the end of time t ,

$$\mu_{i,t} = \Pi_{i,t} = P_t q_{i,t} - C_{i,t}.$$

Firms' decision rules are updated using three genetic operators, *reproduction*, *crossover*, and *mutation*, in the basic GA version or using four genetic operators, reproduction, crossover, mutation, and *election*, in the augmented GA version.

Reproduction makes the copies of individual chromosomes. The criterion used in copying is the value of the fitness function. Chromosomes with higher fitness value are assigned higher probability of contributing an offspring that undergoes further genetic operation. Thus, a probability that a chromosome $A_{i,t}$ will get a copy $C_{i,t}$ is given by

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

The algorithmic form of the reproduction operator is like a biased roulette wheel where each string is allocated a slot sized in proportion to its fitness. A 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 the application of other genetic operators.

Crossover exchanges the 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 $(1, \dots, l - 1)$ and two new strings are formed by swapping the set of binary values to the right of the position k . The total of $n/2$ (n is even integer) pairs are selected and the crossover takes place on each pair with probability $pcross$. An example of the crossover between two strings for $l = 8$ and $k = 4$ is

$$\left\{ \begin{array}{cccc|cccc} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{array} \right.$$

After the crossover is performed, two resulting strings are

$$\left\{ \begin{array}{cccccccc} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \end{array} \right.$$

Mutation is the process of a random change of the value of a position within a string. Each position has a small probability, $pmut$, of being altered by mutation, independent of other positions.

Election tests newly generated offsprings before they are permitted to become members of a new population. The string value of each new offspring, obtained at time t , is decoded in order to obtain the value of the production that an offspring would represent were it used as an actual decision rule. Profit that results from such production decision is computed using the price that prevailed in the market at time $t - 1$, and it represents the offspring's *potential* fitness value. This potential fitness of an offspring is compared to the *actual* fitness values of its parents, i.e., the fitness values of the two parent strings that were evaluated at the end of period $t - 1$. (Parents are the pairs of strings that are taken from the mating pool for the crossover application.)

Possible results of the election operator test are the following: If only one (out of two offsprings for each parent's pair) has a fitness higher than both of its parents, it replaces the parent with lower fitness, while the parent with higher fitness remains in the population. In case that both offsprings have fitnesses higher than the fitness value of each parent, they replace both parents as new members of the population. If both parents have higher fitnesses than their offsprings, they remain in the population of the new generation.

In case of the basic GA, a population of chromosomes that will represent decision rules of the firms at time period t is obtained in the following way: First, the application of the reproduction operator yields a population of n copies. Then, crossover and mutation operators are applied to this population to yield a new population of decision rules that will determine firms' production at t . If the augmented GA is used, the application of reproduction, crossover, and mutation is followed by the application of an election operator.

After the members of the new population are determined, a quantity that will be produced and offered for sale at time t is computed for each firm. Next, individual quantities are summed up and the market price of period t , P_t , is computed using eq. (4).

Costs associated with produced quantities are computed for each firm [eq. (1)], and a profit for firm i (fitness value of a string i), $i = 1, \dots, n$, is calculated using the price of generation t , P_t [eq. (2)].

The above described steps are applied iteratively for T generations. The 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 where the production

decision rules of those firms whose beliefs are given by well-performing strings are copied by others, by virtue of them earning higher profits in the market. Strings with lower fitness values, which means worse production decisions and lower profits, get less copies (or none) in the next generation, as investors or financial intermediaries are not willing to allocate investment funds into an unprofitable production. Crossover and mutation are used to generate new ideas (beliefs) on how much to produce and offer for sale, recombining the existing beliefs and generating new ones with crossover and mutation.

If the 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 fitnesses of these new potential proposals to the old set, under the market conditions observed in the past. Only new ideas that appear promising on such grounds are actually implemented (whereas the generation of new ideas is random, their implementation is not).

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

3.1. Results of simulations

Simulations were conducted for seven different sets of the cobweb model parameter values (see table 1), using both the basic and the augmented algorithm. GA populations consisted of thirty strings and a string length was set to thirty bits. Eight different sets of crossover and mutation rates that were used are given in table 2. Each simulation was conducted for two hundred periods.²

Simulations of the basic GA did not result in convergence to the rational expectations equilibrium values. In every simulation, individual quantities and prices fluctuated for its entire duration. The first set of genetic operator rates ($p_{cross} = 0.6$ and $p_{mut} = 0.0033$) had consistently the smallest magnitude of fluctuations.

On the other hand, simulations which were conducted by using the augmented GA resulted in the convergence of the algorithm to rational expectations

²Multiple runs (using different seed values for the initialization of the random number generator) of simulations were conducted for each combination of the cobweb model parameter values and set of genetic operator rates to ensure that the results are robust to different sequences of random numbers. The same procedure was used for the simulations of the multiple-population GA whose results are reported in the following section. Results of individual runs, not averages over multiple runs, were examined.

Table 1
Parameter values of the cobweb model used in genetic algorithm simulations.

Set	1	2	3	4	5	6 ^a	7 ^a
<i>A</i>	100	10	100	7	1000	2.184	2.296
<i>B</i>	0.02	0.03	0.02	0.003	0.02	0.0152	0.0168
<i>x</i>	3	2	1	2	200	0	0
<i>y</i>	1	1	1	1	1	0.016	0.016

^aSet 6 represents the parameter values of the stable cobweb case, while set 7 represents the values of the unstable cobweb case used in Wellford's experiments.

Table 2
Crossover and mutation rates used in genetic algorithm simulations.

Set	1	2	3	4	5	6	7	8
<i>pcross</i>	0.6	0.6	0.75	0.75	0.9	0.9	0.3	0.3
<i>pmut</i>	0.0033	0.033	0.0033	0.033	0.0033	0.033	0.0033	0.033

Table 3
Results of the simulations for single-population GA (set of cobweb model parameter values).

GA set		1	2	3	4	5	6	7	8
Basic stable	\bar{P}^a	1.174	1.032	1.149	0.973	1.163	1.200	1.108	1.200
	δ	0.081	0.111	0.054	0.073	0.065	0.079	0.059	0.071
	δ_R	0.098	0.141	0.062	0.164	0.077	0.113	0.061	0.106
Basic unstable	\bar{P}	1.210	1.200	1.210	1.180	1.014	1.200	1.630	1.204
	δ	0.066	0.075	0.060	0.116	0.052	0.092	0.062	0.086
	δ_R	0.115	0.108	0.106	0.131	0.118	0.116	0.076	0.120
Augment. stable	\bar{P}	1.119	1.119	1.121	1.121	1.120	1.119	1.120	1.120
	δ	0.005	0.005	0.011	0.011	0.009	0.009	0.011	0.012
	δ_R	0.005	0.005	0.011	0.011	0.009	0.009	0.011	0.012
Augment. unstable	\bar{P}	1.120	1.120	1.121	1.121	1.122	1.121	1.121	1.121
	δ	0.008	0.007	0.012	0.011	0.011	0.011	0.011	0.009
	δ_R	0.008	0.007	0.012	0.012	0.011	0.011	0.011	0.009

^a \bar{P} = average price of a simulation (200 periods), δ = standard deviation about the average price of a simulation, δ_R = standard deviation about the rational expectations equilibrium price.

equilibrium values for all sets of cobweb model parameter values, which include both stable and unstable cases. All the strings in a GA population of each simulation became identical, i.e., the beliefs of all firms about how much to produce and offer for sale converge to the same value which is equal to the optimal quantities when the market price is known.

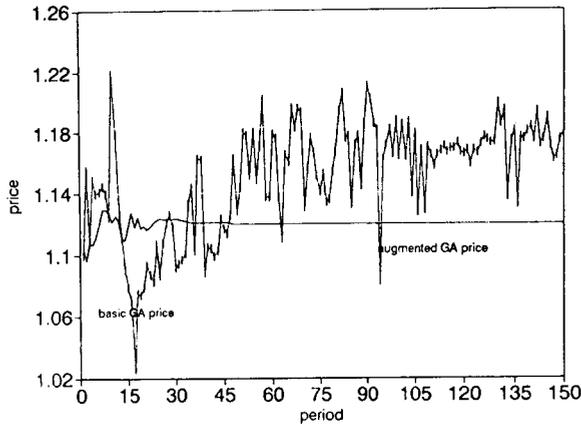


Fig. 1. Single-population basic and augmented GA price patterns – 7th set of the cobweb model parameter values, 1st set of genetic operator rates.

The second set of genetic operator rates ($p_{cross} = 0.6$ and $p_{mut} = 0.033$) resulted in the fastest convergence for all cobweb model parameter values that were examined. Use of other sets of genetic operator rates resulted in greater fluctuations that lasted longer, relative to the results obtained when the second set of rates was used. Average prices, standard deviations about average prices, and standard deviations about rational expectations prices for the sixth and seventh set of the cobweb model parameter values and for all the sets of GA rates that were used are given in table 3.

Price patterns of both the basic and the augmented GA, for the seventh set of the cobweb model parameter values and for the first set of genetic operator rates ($p_{cross} = 0.6$ and $p_{mut} = 0.0033$), are exhibited in fig. 1.

3.2. Discussion of the election operator effect

Necessary condition for the algorithm's convergence to the equilibrium values is that no rule (string) in the GA population deviates from the quantity that maximizes profit at a market clearing price. This also implies that the variance of the population of rules is equal to zero. In equilibrium, the population of identical rules decode to the quantity that equals the value which maximizes profits at a correctly perceived equilibrium price.

While the reproduction operator works towards a reduction in the variance of the rules, mutation works towards maintaining a degree of diversity. In simulations with the basic GA, the extent to which the diversity brought in by mutation offsets the effects of the reproduction operator depends on the rate of mutation and on the complexity of the learning environment. In any event,

continuing effects of mutation will maintain a variance of rules greater than zero as long as a simulation is conducted, resulting in rules which decode to quantities different from the quantities that maximize profits at given prices.

On the other hand, the application of the election operator results in the reduction of rules' deviation from the quantity that maximizes profit at the market clearing price and in the reduction of the population variance over the course of a simulation. Finally, the election operator brings the variance of the population rules to zero as the algorithm converges to the equilibrium values.

Once the algorithm converges to the equilibrium values, all strings become identical and, thus, all are decoded into the same equilibrium value of production, q^* . Mutation will continue to generate new, different strings at a rate that is exogenously given, but none of these strings will be accepted into the population. Any string that decodes into the quantity q such that $q \neq q^*$ will have lower fitness value, evaluated at the equilibrium price P^* . Thus, the election operator enables an endogenous shut-off of mutation, resulting in its effective rate equal to zero in equilibrium.

This does not, however, mean that the adjustment of the algorithm to new equilibrium values will be prevented if the underlying parameters of the economy change (for example, a change in the production function or a change in the demand schedule). Suppose that for a given set of the cobweb model parameters, x , y , A , and B , the GA converges to the equilibrium with the equilibrium price P^* and equilibrium individual quantity q^* . All strings are identical and decode to this quantity. Then, at time period $T + 1$, a parameter of the demand schedule changes from the value A to the value A' . This will also result in the change of the equilibrium values of price and individual quantity to the new values of $P^{*'} and $q^{*'}$.$

When genetic operators are applied to yield a population of production decision rules at $T + 1$, reproduction and crossover will have no effect on the population of identical strings. Mutation may generate some new strings with quantities different from q^* , but they will not be accepted into the population of period $T + 1$ since they are evaluated at the price $P_T = P^*$. Thus, at the period when the change occurs, $T + 1$, all strings that determine market supply will still decode to the quantity q^* . However, these strategies will no longer result in price P^* . The price at $T + 1$ will be given by

$$P_{T+1} = A' - Bnq^* \neq P^* .$$

At time $T + 2$, after the application of reproduction, crossover, and mutation, strings are evaluated at price P_{T+1} and any string that decodes to quantity $q_i \neq q^*$, which has a higher fitness value (that is now possible since q^* is no longer optimal quantity), will be accepted into the population to give decision rules for production at $T + 2$. If such a string (or strings) is not created at $T + 2$,

it is created in one of the subsequent periods. This way diversity is brought into the population and reproduction, crossover, and mutation will work their way through towards the adjustment to the new equilibrium values. [For a detailed discussion of the behavior of the GA with an election operator in the environments in which a parameter of the economic model changes see Arifovic (1991).] In computer simulations, the adjustment of GA populations of identical strings to new equilibrium values, when the change in a parameter occurs, does not last longer than the adjustment of initial GA populations of strings that are randomly generated.³

It should be noted that the information about the economic environment available to cobweb firms that use the augmented GA is the same as the information available to firms that use the basic GA. The application of the election operator requires only the use of the previous period market price to test newly generated strings. The market price is also observed and used by basic GA firms to compute the profits they earn, i.e., the fitnesses of the decision rules they used.

4. Multiple-population genetic algorithm

An alternative way to represent economic agents' learning by means of the GA is to endow every individual agent with a whole population of strings. Then we can think of these strings in such a population as being an agent's mutually competing ideas about what his behavior in a given environment should be. In each time period, only one string is selected as a string that determines the 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 the whole collection, he still evaluates *all* of his alternative ideas, upon obtaining the information about variables whose values he did not know at the time of his decision making. This way the agent gets the information as to what the value of his objective function would have been had he used a particular string from the population of competing decision variable values. The agent uses this information in the process of updating his beliefs (updating is performed using genetic operators) to assign higher probability of reproduction to those strings that yielded higher values in the performance evaluation.

³Another possible way to deal with the problem of the mutation rate effects which prevent the algorithm's convergence is to use an exponentially decaying mutation rate which is used in simulated annealing and in some genetic algorithm applications. The rate at which mutation decreases as the number of simulation periods increases is given exogenously. While the inclusion of the election operator does not impair the system's capability for new adjustment in case of change in underlying parameters of the model, this ability does get reduced if the mutation rate is exogenously decreased over time [see Arifovic (1992)].

The multiple-population GA is applied to the cobweb model described in the previous section. It is richer compared to the single-population GA in the sense that we can now think of firms as having a number of different ideas about their possible production quantities, which they evaluate and then choose one among them, using some adopted selection rule. On the other hand, it still does not require any excessive computational capacity on the part of the agents. Also, as in the case of the single-population GA, there is no assumption of profit maximization on the part of firms preceding their production decision. At the same time, it is not *simplicistic* in the way of the single-population GA, where each agent has only one strategy at his disposal.

4.1. Description of the model

Each firm j ($j = 1, \dots, m$) has a collection A_t^j of n possible decision rules which is given as a population of binary strings. String i , member of the j th firm's collection at generation t , is denoted by $A_{i,t}^j$. The decoded and normalized value of each string represents a possible value of a next-period production, $q_{i,t}^j$. A string i ($i = 1, \dots, n$) is assigned a *competition* fitness which is equal to the profit at the price that prevailed at generation $t - 1$ and the quantity represented by string i .

Every string i in a firm j 's population of strings has a chance to be selected as the string that will supply the value representing the quantity of firm j 's next-period production. A probability that a particular string is chosen as firm j 's decision rule for production at t is proportional to its competition fitness and is given by

$$\pi_{i,t}^j = \mu_{i,t}^{j,c}(P_{t-1}) \left/ \sum_{i=1}^n \mu_{i,t}^{j,c}(P_{t-1}) \right.,$$

where $\pi_{i,t}^j$ is the probability that a string i , a member of firm j 's collection of possible decision rules at generation t , is chosen and $\mu_{i,t}^{j,c}(P_{t-1})$ is the competition fitness of a string i evaluated at the price that prevailed in the market at generation $t - 1$. The selected string becomes firms j 's *realized* decision rule that gives quantity $q_t^{r,j}$ as the quantity to be produced in time period t .⁴

The market clearing price is given by

$$P_t = A - B \sum_{j=1}^m q_t^{r,j}. \quad (6)$$

⁴The information that firms have in this setup is the same as in the single-population framework. They know only the previous-period price and their own profits earned. They do not know the quantities supplied by other firms. The assumption of the original cobweb model that a firm believes its own quantity is not going to affect the market is maintained here as well. It is also the assumption of all adaptive algorithms described in the second section.

Firm j 's cost of producing a quantity $q_t^{r,j}$, represented by a chosen string, at generation t is given by

$$C_t^j = xq_t^{r,j} + \frac{1}{2}y(q_t^{r,j})^2, \quad (7)$$

and its profit at t is given by

$$\Pi_t^j = P_t q_t^{r,j} - C_t^j(q_t^{r,j}). \quad (8)$$

Price P_t is used to determine *reproduction* fitness values of all strings in all m populations. This fitness is equal to the profit at the price that cleared the market at t and at the quantity represented by a particular string:

$$\mu_{i,t}^j = \Pi_{i,t}^j = P_t q_{i,t}^j - C_{i,t}^j,$$

where $\mu_{i,t}^j$ is a reproduction fitness value⁵ of a string i at t in a population that belongs to the j th firm, $\Pi_{i,t}^j$ is a profit that a firm j would have earned had string i from its population been used for production, $q_{i,t}^j$ is a quantity represented by string i , and $C_{i,t}^j$ are the costs associated with the production of that particular quantity.

After the evaluation of their fitness takes place, operators of the basic GA, reproduction, crossover, and mutation, are applied within each firm's collection of strings to yield the new sets of possible decision rules for production at $t + 1$.

Decoded and normalized values of the members of a collection A_{t+1}^j give quantities $q_{i,t+1}^j$ ($i = 1, \dots, n$), which are possible quantities for the production at $t + 1$. In order to reach a decision about the production at $t + 1$, each firm j computes competition fitnesses of the newly generated strings in a collection A_{t+1}^j , using price P_t , and chooses a single string which will give the quantity to be actually produced.

The process is repeated for T generations. Initially, all collections of strings are randomly generated.

In the augmented multiple-population GA version, the election operator is added. Before entering the competition for a string that determines next period's production, strings that are generated at period $t + 1$ as a result of application of genetic operators have to pass a *qualifying* test given by the election operator. Within each collection of strings, the election operator is applied in the same manner as was described in case of the single-population GA.

⁵Notice that this is a fitness that is computed after market clearing takes place and price P_t is known. It is used when the reproduction operator is applied to determine the probability that a string i is reproduced. On the other hand, the competition fitness value is determined after new strings have been created through the action of genetic operators and it is used to determine the probability that a particular string is chosen as the actual decision rule.

4.2. Results of simulations

Simulations were conducted for the number of firms $m = 2, m = 3, m = 5,$ and $m = 6$ and for the parameters values of the cobweb model reported in table 1 (the same that were used for the single-population GA application). For each of these specifications, both the basic and the augmented GA were applied, using the sets of values of the genetic operator rates given in table 2.

The results of the simulations show that the augmented GA converged to the rational expectations equilibrium values for all sets of cobweb model parameter values, including stable and unstable case, and for all sets of genetic operators. At the same time, the basic GA kept oscillating until the end of every simulation. These oscillations are the result of the effects of mutation which were discussed in the previous section. No significant decrease in the magnitude of oscillations was observed when simulations were conducted with a low rate of mutation (0.0033) and for a large number of periods (10,000). Table 4 contains average prices, standard deviations about average prices, and standard deviations about rational expectations prices for the sixth and seventh set of the cobweb model parameter values and for all sets of the GA rates. Results for both the basic multiple-population GA ($m = 5$) and the augmented multiple-population GA ($m = 5$) are included.

Although only a single string determines a firm's next-period production, by the time the augmented GA achieves convergence to the rational expectations equilibrium, the entire firm's population of strings converges to the same value.

Table 4

Results of the simulations for the multiple-population GA (set 6 and 7 of cobweb model parameter values).

GA set		1	2	3	4	5	6	7	8
Basic stable	\bar{P}^a	1.109	1.230	1.121	1.138	1.114	1.127	1.117	1.139
	δ	0.065	0.111	0.060	0.102	0.078	0.109	0.079	0.113
	δ_R	0.066	0.111	0.060	0.104	0.078	0.109	0.079	0.115
Basic unstable	\bar{P}	1.135	1.131	1.124	1.119	1.163	1.134	1.168	1.126
	δ	0.062	0.099	0.061	0.110	0.055	0.112	0.074	0.100
	δ_R	0.063	0.099	0.061	0.110	0.070	0.113	0.088	0.100
Augment. stable	\bar{P}	1.115	1.120	1.120	1.120	1.121	1.118	1.113	1.113
	δ	0.006	0.006	0.012	0.010	0.010	0.010	0.024	0.022
	δ_R	0.008	0.007	0.012	0.010	0.010	0.010	0.025	0.023
Augment. unstable	\bar{P}	1.120	1.120	1.120	1.121	1.117	1.120	1.110	1.110
	δ	0.010	0.007	0.013	0.013	0.011	0.012	0.025	0.025
	δ_R	0.010	0.007	0.013	0.013	0.011	0.012	0.027	0.027

^a \bar{P} = average price of a simulation (200 periods), δ = standard deviation about the average price of a simulation, δ_R = standard deviation about the rational expectations equilibrium price.

By contrast, at the end of each simulation, there is still a difference between the strings of a population that belongs to a firm which uses the basic GA.

Fig. 2 shows the price level behavior for the simulation of the multiple-population basic GA ($m = 2$, sixth set of the cobweb model parameter values, second set of genetic operators rates), while fig. 3 shows the behavior of the same variable for the simulation of the multiple-population augmented GA ($m = 2$, seventh set of the cobweb model parameter values, second set of genetic operator rates).

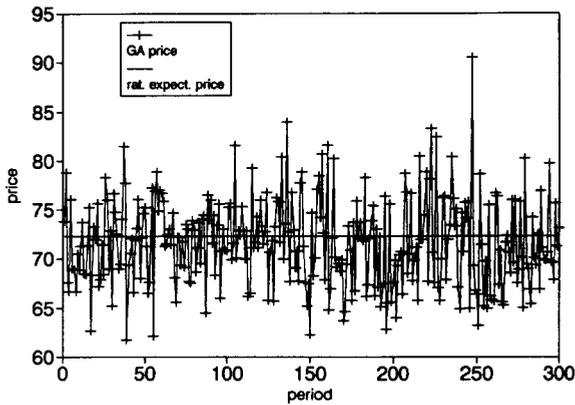


Fig. 2. Multiple-population basic GA – price and rational expectations price, $m = 2$, 6th set of the cobweb model parameter values, 2nd set of genetic operator rates.

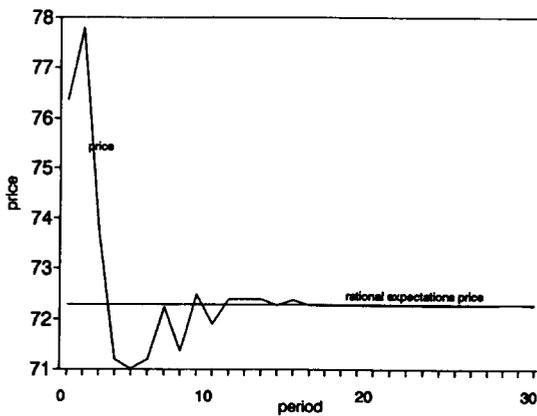


Fig. 3. Multiple-population augmented GA – price and rational expectations price, $m = 2$, 7th set of the cobweb model parameter values, 2nd set of genetic operator rates.

5. Comparison with experimental results

Three aspects of Wellford's experimental data, namely the absence of divergent patterns in the cobweb unstable case, fluctuations around the cobweb model equilibrium values, and the greater price variance of the unstable case, will be used for the evaluation of the performance of cobweb expectations, sample average of past prices, least squares, and the GA.⁶ Figs. 4 and 5 exhibit the behavior of average individual quantities with plus and minus one standard deviation, using data from Wellford's experiments 1 (stable case) and 3 (unstable case).

Fluctuations of prices and quantities around their rational expectations values that characterize models in which agents form cobweb expectations are observed in Wellford's experiments as well, but (as was already discussed) for the parameter values of the unstable case Wellford's experiments did not follow the divergent price path predicted by the model in which agents have cobweb expectations. Fig. 6 exhibits the pattern of the individual quantity for the cobweb stable case.

The adaptive scheme in which firms make their price forecast by taking the sample average of past prices converges for both stable and unstable case, but

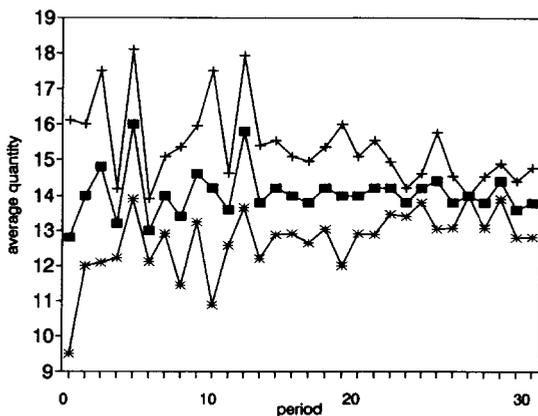


Fig. 4. Wellford's experiment 1 (stable case) – average quantity, average quantity plus one standard deviation, average quantity minus one standard deviation.

⁶The results of Wellford's experiments conducted under conditions of incomplete information were those that could be used for the comparison. In the experiments with incomplete information, sellers received information only about the last-period price and about their own earned profits, but no information about the total quantity sold or about quantity choices made by other market participants. This corresponds to the GA environment in which strings do not obtain any other information except for their fitness (profit) and the price of the last period. That is also the only information available to agents who adapt by using cobweb expectations, sample average of past prices, or least squares.

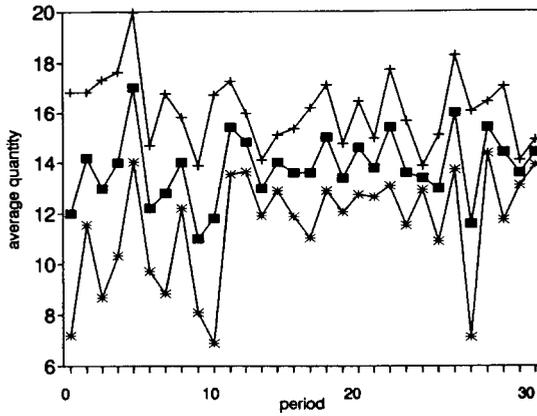


Fig. 5. Wellford's experiment 3 (unstable case) – average quantity, average quantity plus one standard deviation, average quantity minus one standard deviation.

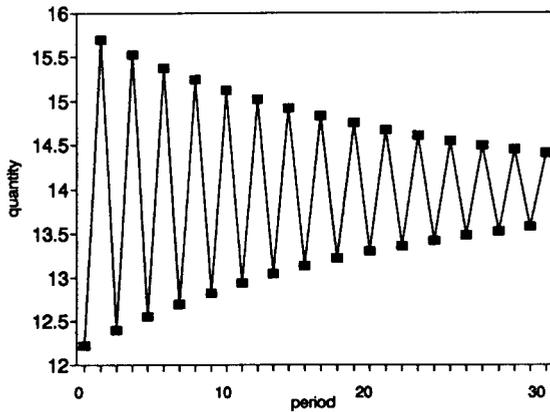


Fig. 6. Cobweb expectations learning scheme (stable case) – individual quantity.

since its convergence is smooth, it does not bear resemblance to the experimental data which is characterized by fluctuations around equilibrium values. Moreover, the price variance of the stable case is not greater than the price variance of the unstable case. The behavior of this algorithm in computer simulation for Wellford's unstable set of parameter values is given in fig. 7.

For the parameter values of the cobweb stable model, the least squares algorithm exhibits the price and quantity fluctuations that die out as the algorithm converges to rational expectations values. On the other hand, it does not converge in computer simulations for the unstable cobweb case. The price and quantity fluctuations become larger and larger as the algorithm diverges

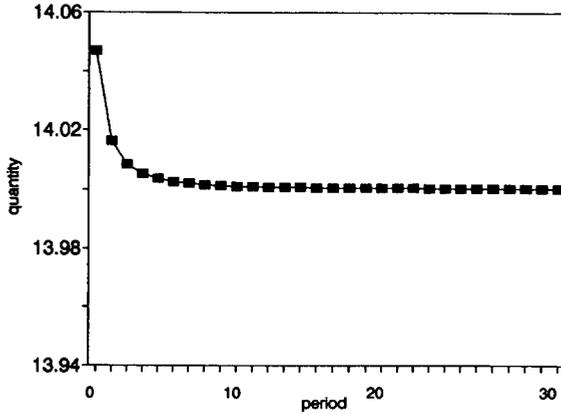


Fig. 7. Sample average of past-price learning scheme (stable case) – individual quantity.

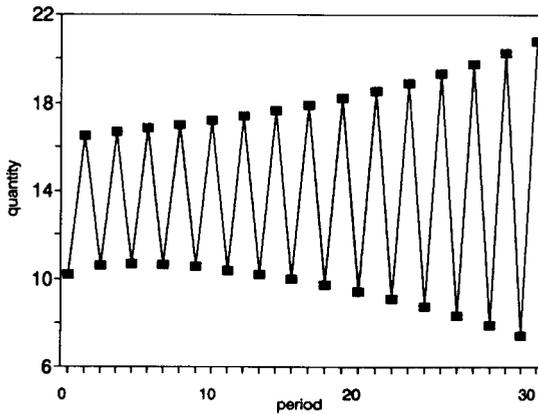


Fig. 8. Least squares learning scheme (unstable case) – individual quantity.

away from the rational expectations values. Fig. 8 shows the pattern of the individual quantity for thirty periods, for the parameters of Wellford's unstable case.

Both single-population and multiple-population augmented GA's converge for the stable and the unstable cobweb case. Prior to convergence, the GA exhibits fluctuations around the equilibrium values. The patterns of individual quantities with plus and minus one standard deviation, generated by the multiple-population augmented GA for the cobweb stable and cobweb unstable case, are presented in figs. 9 (stable) and 10 (unstable).

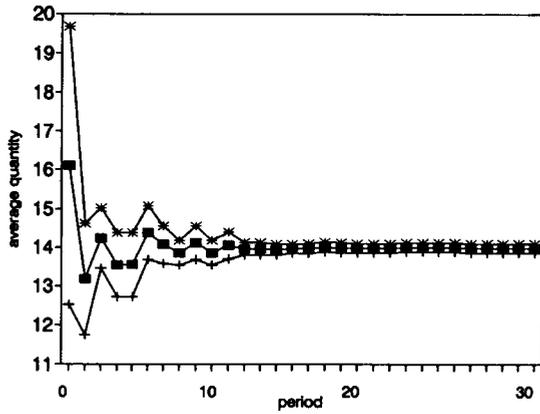


Fig. 9. Multiple-population augmented GA – average quantity, average quantity plus one standard deviation, average quantity minus one standard deviation, 6th set of the cobweb model parameter values, 2nd set of GA rates.

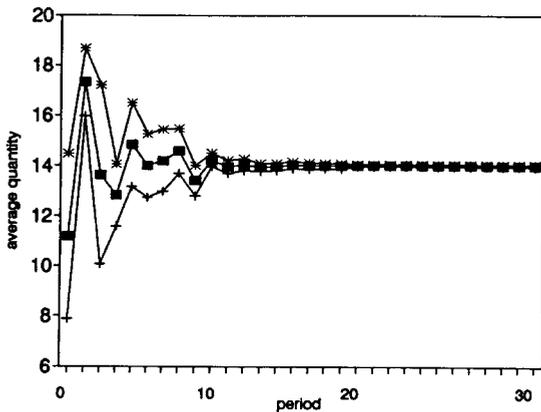


Fig. 10. Multiple-population augmented GA – average quantity, average quantity plus one standard deviation, average quantity minus one standard deviation, 7th set of the cobweb model parameter values, 4th set of GA rates.

GA price patterns also show that the variance of unstable cases is greater than that of stable ones. (Description of the variance testing procedure is given in appendix 2.) Variance tests were performed on data generated using both the single-population and multiple-population GA. For each of these tests, the hypothesis that the price variances of both stable and unstable case are the same was rejected in favor of the hypothesis that the variance of the cobweb unstable case is greater than that of the cobweb stable case.

7. Conclusion

Two GA cobweb model designs were developed, the single-population and the multiple-population GA. In the single-population GA design, each firm has only one string which it uses for its decision making. In the multiple-population GA design, each firm is endowed with a population of strings, a collection of competing rules, for its decision making. In every time period, based on the rules' previous-period performance, a firm chooses one among them as its actual decision about how much to produce in that period. The mechanism for a choice of a particular string as the actual decision rule resembles the mechanisms employed in parallel algorithms that are used in models of human cognition.⁷ A common feature is the competition of alternative rules in which the probability of winning the competition is determined on the basis of a rule's past performance in a given environment.

Two versions of the GA, basic and augmented, were used in computer simulations of both GA designs. Besides reproduction, crossover, and mutation, which are the operators of the basic GA, the augmented version also includes an election operator. This operator has been designed and added to the algorithm in order to overcome difficulties related to the way mutation influences the convergence process.

The single-population augmented GA and the multiple-population augmented GA converged in simulations to rational expectations equilibrium values for the cobweb stable and unstable case. Both single-population basic GA and multiple-population basic GA exhibited fluctuations, without settling to rational expectations equilibrium values.

The results of the cobweb experimental economies were compared to the behavior of the GA and of three other learning algorithms (least squares learning algorithm, algorithm in which agents form cobweb expectations, sample average of past prices). The experimental economies do not exhibit divergent patterns for those parameter values for which the cobweb expectations and the least squares, for example, diverge and for which, at the same time, GA converges. The sample average of the past-price adaptive scheme is characterized by the smooth convergence to rational expectations equilibrium values for the cobweb stable case and the cobweb unstable case. Thus, it does not capture fluctuations around the equilibrium values observed in the experiments. At the same time, the GA exhibits fluctuations prior to convergence. Since GA price variance of the cobweb unstable case is greater than the price variance of the stable case, the GA captures well the same feature observed in the experimental data.

In general, when compared to the other algorithms that were used in modeling of learning in the cobweb environment, the GA requires less prior

⁷See, for example, Rumelhart, McClelland, and the PDP Research Group (1987), Holland et al. (1986), and Edelman (1987).

competence in the specific task facing the agents. Most of these schemes assume that agents know how to maximize their objective function, whereas such an assumption is not required within the GA framework. Instead, while learning about their environment, agents also *learn* how to maximize their objective functions.

This feature of the GA behavior is similar to the characteristics of human behavior observed in experiments. On one hand, choice surveys and informal subject debriefing suggest that people do not solve their decision problems by applying logic and techniques used by economists to solve the same models. On the other hand, behavior observed in experimental economies (supply and demand, oligopoly, bargaining, etc.) is often consistent with the predictions of market theories. As Smith (1989) suggests, changes in the reference frame or norms of behavior over time are induced by the invisible reality of entry or exit opportunity costs and therefore observed adaptation may be imposed upon agents who need not have a cognitive grasp of the causes that are driving changes.

By the same token, economies in which agents that use the GA to update their beliefs start out with a variety of initial beliefs, but those beliefs that cannot withstand the competition of better performing rules are eliminated from the population. Agents do not have a cognitive grasp of the causes that are driving changes. The whole adaptation is based on the ability to generate new beliefs randomly and on the survival of the fittest among competing rules.

Appendix 1

At each period t , firm i ($i = 1, \dots, n$) chooses an action $q_{i,t} \in [0, \bar{q}]$. The $q_{i,t}$ is approximated by an l -length string written over binary alphabet $\{0, 1\}$. For example, if $l = 3$, then the choice is given by one of eight possible binary strings with the following interpretation:

Action:	Interpretation
0 0 0:	$q_{i,t} = 0$
1 0 0:	$q_{i,t} = 1 \times (\bar{q}/7)$
0 1 0:	$q_{i,t} = 2 \times (\bar{q}/7)$
0 1 0:	$q_{i,t} = 3 \times (\bar{q}/7)$
0 0 1:	$q_{i,t} = 4 \times (\bar{q}/7)$
1 0 1:	$q_{i,t} = 5 \times (\bar{q}/7)$
0 1 1:	$q_{i,t} = 6 \times (\bar{q}/7)$
1 1 1:	$q_{i,t} = 7 \times (\bar{q}/7)$

As l increases, a finer partition of the choice set is allowed.

Appendix 2

If one wants to test the same hypothesis that Wellford tested on her experimental data (i.e., that the price variance of the unstable cobweb treatments is equal to that of the stable treatments) on data generated by the GA, two different approaches can be taken.

One approach is to assume that agents use the same set of values of crossover and mutation rates in all experiments, and for both stable and unstable cobweb case. With this assumption, data used for the hypothesis testing is generated by a single set of genetic operator values. The alternative approach is to assume that agents use different sets of values in different experiments.

Besides the fact that the price variance observed in cobweb stable experiments was lower overall than the price variance in the unstable cases, there were also substantial differences in the price variance across experiments of the same cobweb type. In some of the cobweb stable experiments, the price remained close to the equilibrium price throughout the whole experiment with the resulting low price variance. In others, it exhibited substantial fluctuations. This was observed in the cobweb unstable experiments as well.

On the other hand, in GA simulations, different sets of genetic operator values generate different values of price variance. Higher rates of crossover and mutation generate, on average, higher price variance, while lower genetic operator rates result in lower price variance. Thus, a set of observations used for the computations of variance in price of each cobweb type should consist of data generated with the whole range of values of crossover and mutation rates.

For the first approach, data was generated conducting 20 iterations for each set of genetic operator values. Each iteration lasted for 30 generations and was initialized with a different seed for the random number generator. All sets of values were implemented for both stable and unstable cobweb case (20 iterations for each). The same sequence of seed numbers was used for each set of values.

The null hypothesis that the variance in price across all periods for unstable cobweb type for a single set of genetic operator rates is equal to that of stable cobweb type for the same set of crossover and mutation rates was tested for each of the above specified set of values, at 5% significance level. It was rejected in favor of the hypothesis that the variance in price is greater in the unstable than in the stable case for each set of values.

Data for the second approach to hypothesis testing was generated conducting one iteration for each set of the genetic operator values. Each iteration lasted for 30 generations. The whole simulation consisted of 20 iterations with 20 different sets of values. One simulation was performed for the cobweb stable case and one for the unstable case, using the same sequence of genetic operator rates for both cases. Data generated in a single simulation for each cobweb type represented the basis for the calculation of the respective price variances. Crossover rates varied from 0.6 to 0.9 and were combined with mutation rate values of 0.0033

and 0.033. The hypothesis that the price variances of the stable and unstable case are the same was rejected again, at 5% significance level, in favor of the hypothesis that the variance of the cobweb unstable case is greater than that of the cobweb stable case.

References

- Arifovic, J., 1991, Learning by genetic algorithms in economic environments, Doctoral dissertation (University of Chicago, Chicago, IL).
- Arifovic, J., 1992, Adaptation of genetic algorithm in the environments with changing parameters, Unpublished manuscript (McGill University, Montréal).
- Arthur, B., 1991, Designing economic agents that act like human agents: A behavioral approach to bounded rationality, *American Economic Review: Papers and Proceedings of the 103rd Annual Meeting of the American Economic Association*, 353–350.
- Binmore, K. and L. Samuelson, 1990, Evolutionary stability in repeated games played by finite automata, Working paper (University of Michigan, Ann Arbor, MI and University of Wisconsin, Madison, WI).
- Blume, L.M., M.M. Bray, and D. Easley, 1982, Introduction to the stability of rational expectations equilibrium, *Journal of Economic Theory* 26, 313–317.
- Brandenburger, A., 1984, Information and learning in market games, Unpublished manuscript (Stanford University, Stanford, CA).
- Bray, M.M. and N.E. Savin, 1986, Rational expectations equilibria, learning, and model specification, *Econometrica* 54, 1129–1160.
- Carlson, J., 1969, An invariably stable cobweb model, *Review of Economic Studies* 69, 360–362.
- Crawford, V.P., 1989, An 'evolutionary' explanation of Van Huyck, Battalio, and Beil's experimental results on coordination, Working paper (University of California, San Diego, CA).
- DeCanio, S.J., 1979, Rational expectations and learning from experience, *Quarterly Journal of Economics* 93, 47–57.
- Edelman, G.M., 1987, *Neural Darwinism* (Basic Books, New York, NY).
- Ezekiel, M., 1938, The cobweb theorem, *Quarterly Journal of Economics* 52, 255–280.
- Frydman, R., 1982, Towards an understanding of market processes, *American Economic Review* 72, 652–668.
- Goldberg, D.E., 1989, *Genetic algorithms in search, optimization and machine learning* (Addison-Wesley, New York, NY).
- Holland, J.H., 1970a, Robust algorithms for adaptation set in a general formal framework, *Proceedings of the IEEE Symposium on Adaptive Processes – Decision and Control* 17, 5.1–5.5.
- Holland, J.H., 1970b, Processing and processors for schemata, in: E.L. Jacks, ed., *Associative information processing* (American Elsevier, New York, NY).
- Holland, J.H., 1975, *Adaptation in natural and artificial systems* (University of Michigan Press, Ann Arbor, MI).
- Holland, J.H. and J.H. Miller, 1991, Artificial adaptive agents in economic theory, *American Economic Review: Papers and Proceedings of the 103rd Annual Meeting of the American Economic Association*, 365–370.
- Holland, J.H., K.J., Holyoak, R.E. Nisbett, and P.R. Thagard, 1986, *Induction – Processes of inference, learning and discovery* (MIT Press, Cambridge, MA).
- Holt, C. and A. Williamil, 1986, A laboratory experiment with a single-person cobweb, *Atlantic Economic Journal* 219, 51–54.
- Lucas, R.E., Jr., 1986, Adaptive behavior and economic theory, *Journal of Business* 59, 401–426.
- Marcet, A. and T.J. Sargent, 1987, Convergence of least squares learning in environments with hidden state variables and private information, *Journal of Political Economy* 97, 337–368.
- Marimon, R., E. McGrattan, and 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.
- Miller, J.H., 1986, A genetic model of adaptive economic behavior, Working paper (University of Michigan, Ann Arbor, MI).

- Miller, J.H., 1989, The coevolution of automata in the repeated prisoner's dilemma, Working paper 89-003 (Santa Fe Institute, Santa Fe, NM).
- Miller, J.H. and J. Andreoni, 1990a, A coevolutionary model of free riding behavior: Replicator dynamics as an explanation of the experimental results, Working paper (Santa Fe Institute, Santa Fe, NM and University of Wisconsin, Madison, WI).
- Miller, J.H. and J. Andreoni, 1990b, Auctions with adaptive artificially intelligent agents, Working paper 90-01-004 (Santa Fe Institute, Santa Fe, NM).
- Muth, J.F., 1961, Rational expectations and the theory of price movements, *Econometrica* 29, 315-335.
- Nerlove, M., 1958, Adaptive expectations and cobweb phenomena, *Quarterly Journal of Economics* 72, 227-240.
- Nyarko, Y., 1990, Bayesian rationality and learning without common priors, Unpublished manuscript (New York University, New York, NY).
- Rumelhart, D.E., J.L. McClelland, and the PDP Research Group, 1987, *Parallel distributed processing* (MIT Press, Cambridge, MA).
- Rust, J., R. Palmer, and J.H. Miller, 1989, A double auction market for computerized traders, Working paper 89-001 (Santa Fe Institute, Santa Fe NM).
- Smith, V.L., 1989, Theory, experiment and economics, *Journal of Economic Perspectives* 3, 151-169.
- Townsend, R.M., 1978, Market anticipations, rational expectations, and Bayesian analysis, *International Economic Review* 19, 481-494.
- Wellford, C.P., 1989, A laboratory analysis of price dynamics and expectations in the cobweb model, Discussion paper 89-15 (University of Arizona, Tuscon, AZ).