Learning and Behavioral Stability — An Economic Interpretation of Genetic Algorithms

Thomas Riechmann*

July 22, 1998

Abstract

This article tries to connect two separate strands of literature concerning genetic algorithms. On the one hand, extensive research took place in mathematics and closely related sciences in order to find out more about the properties of genetic algorithms as stochastic processes. On the other hand, recent economic literature uses genetic algorithms as a metaphor for social learning. This paper will face the question what an economist can learn from the mathematical branch of research, especially concerning the convergence and stability properties of the genetic algorithm.

It is shown that genetic algorithm learning is a compound of three different learning schemes. First, every particular scheme is analyzed. Then it will be pointed out that it is the combination of the three schemes that gives genetic algorithm learning its special flair: A kind of stability somewhere in between asymptotic convergence and explosion.

1 Introduction

As a consequence of the discussion concerning the concepts of perfect and bounded rationality various suggestions have been made which learning mechanism to use instead of letting economic agents know everything they need to know in order to solve their economic problems.[1]

One of the metaphors that can be used for economic learning is that of genetic algorithm learning (GA learning). Genetic algorithm learning is a way of social rather than individual learning.

*Universität Hannover, Institut für Volkswirtschaftslehre

[1]See, for example Sargent (1993).

Social learning always means learning from each other. Thus, there simply is no GA learning by single, isolated agents. This fact suggests a close connection of GA learning to evolutionary economic theory, both fields of research heavily relying on a population concept instead of focusing an economic agent as an isolated individual. The fact that there is no isolated genetic learning also reveals a significant conceptual difference between GA learning on the one hand and statistical learning mechanisms (see e.g. Lucas (1986) or Marcet and Sargent (1989)) or other forms of artificial intelligence based learning (see e.g. Heinemann (1998) for neural network learning) on the other.

Social learning as gathering information according to the simple rules of genetic algorithms means in fact learning according to three different learning techniques: learning by imitation (selection/reproduction), learning by communication (crossover) and learning by experimentation (mutation). Recent economic research shows that genetic algorithm learning performs quite well as a learning mechanism when applied to some standard benchmark cases of economic theory (Andreoni and Miller (1995), Arifovic (1994, 1995, 1996), Bullard and Duffy (1998), Dawid (1996a). Genetic algorithm learning is able to reproduce the results of at least some mainstream economic models, especially concerning their stability properties. One of the scientific challenges to GA learning research is to find out if there are certain properties of genetic algorithms which lead genetic learning models at least to the neighborhood of the results of mainstream economic models. If such properties are found, GA learning could (among other things) serve as a behavioral foundation of mainstream economics, supporting it at its Achilles' heel: its problems at the field of heterogeneity and the interaction of diverse economic agents.[2]

Thus, the aim of this paper is to determine, which are the properties of the GA that lead to the observed similarities of the results of genetic algorithm learning and the outcome of analytical models. Accordingly, the main question to be answered within this paper is: 'Does genetic algorithm learning lead to behavioral stability, and if so, how?'

Economic models using the metaphor of genetic algorithm learning have been widely employed.[3] Although there are quite a few articles about the properties of genetic algorithms in more mathematical fields of research (Davis and Principe (1993), Nix and Vose (1992), Rudolph (1994), Goldberg and Segrest (1987)), there is a lack of theoretical work describing the basic properties of genetic algorithms *used in economic research*.[4] More than this, there is a large amount of work ad-

[2]GA learning models can do even more than just supplement mainstream economics. A first step beyond the abilities of mainstream economic theory is Arifovic (1995), who uses GA learning as a tool of equilibrium selection.

[3]For a survey, see Clemens and Riechmann (1996).

[4]There is at least some work by Dawid (1996a, b), who tries to cope with this problem.

dressing the dynamics of populations of biological or artificial entities within a selective environment, which could be used for the analysis of economic genetic algorithms (e.g. Maynard Smith (1982), Hofbauer and Sigmund (1988), Weibull (1995), Riechmann (1998)).

In order to examine genetic algorithm learning, this article applies the results of these rather technical papers to the field of economic theory, putting its emphasis mainly on the interpretation of the results and to a smaller extent on the mathematical techniques of obtaining them.

The paper starts by briefly reviewing concepts of dynamics and stability (section 2). Then, a mathematical model of a genetic algorithm is set up and analyzed. After that, the result of the analysis is interpreted in terms of learning behavior in a market economy. The paper ends with a summary.

2 Stability

Following mainstream economic literature, a large number of economic problems result in situations which can be described as some kind of stable equilibria. Stability in mainstream economics describes a situation where — after some stage of transitional dynamics — a state is established in which decisions of the economic agents cease to change (asymptotic stability), change within a restricted space of alternatives (Ljapunov–stability) or change within some regular manner (cyclical stability).[5]

A closer look reveals that these notions of stability are mainly notions of macro stability. The state of a society in its role as an economic aggregate can be described by macro data such as equilibrium prices and quantities. Mainstream economic dynamics solely focuses the movement of macro data. Moreover, mainstream economics has no real autonomous concept of micro dynamics, i.e. dynamics of individual behavior. A distinct notion of micro behavior simply is not needed. As long as economic research relies on the concept of the 'representative individual'[6], micro and macro dynamics are just the same.

Economic models which give up reliance on the concept of the representative agent and use a more explicit formulation of heterogeneity may cause the notions of macro and micro dynamics to fall apart.

GA learning models demonstrate various combinations of macro and micro level dynamics. Some examples are given in the following:

[5]These rather naive description can easily be refined by consulting a textbook on mathematical or economic dynamics, such as Azariadis (1993) or Hofbauer and Sigmund (1988).

[6]For a contrast, see Franke (1997), who develops a concept of a representative individual explicitly basing on notions of heterogeneity.

- Aggregate data remains unchanged in time ('asymptotic stability') and individual behavior is identical for all economic agents. This result has been gained by e.g. Arifovic's (1994) augmented GA.

- Aggregate data remains unchanged in time ('asymptotic stability') while individual behavior is heterogenous and regularly changing from time to time.[7] Dawid (1996a) finds stability of this type as a possible result of his extended cobweb model.

- Aggregate data comes from a finite set of numbers ('Ljapunov stability') while individual behavior is heterogenous and changing in such a way that only a finite number of social behavioral patterns will show up. This is the kind of dynamics most GA learning models lead to, e.g. Arifovic's (1994) basic GA.

- Aggregate data changes regularly in cycles ('cyclical stability') while different agents behave differently in the same period, and every agent changes his behavior from time to time. This kind of dynamics can be found in GA learning models like Riechmann (1997).

Throughout the rest of this paper, dynamics of the different learning methods that are part of the GA learning process will be characterized by both, their macro and their micro behavior.[8]

3 Genetic Algorithm Learning as a Markov Process

3.1 The Basics

Standard genetic algorithms are (computational) algorithms which transfer a set of genetic individuals from one generation to the next. Genetic individuals are coded as strings of (binary) bits. The set of genetic individuals of the same generation is called a genetic population. In the standard genetic algorithm, each population is subject to four genetic operators, selection, reproduction, recombination and mutation, to finally result in the next population.[9]

[7] As an illustration, imagine a market in which two quantity decisions are equally attractive to the suppliers: 100 and 0. A situation arises in which in every period 30% of the suppliers choose 100 and 70% choose 0. Every time an 0–supplier changes his quantity to 100 a different supplier changes his from 100 to 0. Thus, the macro–level is stable whereas the micro level is not.

[8] See Björnerstedt and Weibull (1996) for a similar approach to learning mechanisms.

[9] Meanwhile, there are various textbooks and articles describing the standard genetic algorithm in detail. One of the first, yet still one of the best may be Goldberg (1989). More good ones are Davis (1991) and Michell (1996).

A genetic individual of length L consists of L symbols '1' and '0', so that S, the set of all possible different genetic individuals of length L is given as

$$S \in \{0, 1\}^L \tag{1}$$

From that it is clear that there are

$$|S| \equiv N = 2^L \tag{2}$$

different genetic individuals or genetic individuals of a different type.

The 'value' of a genetic individual is obtained by decoding the bit string.[10] In economic models this value describes the behavioral strategy[11] of an economic agent, e.g. the quantity decision of a supplier in a cobweb model (Arifovic (1994)). According to (2) there are N different strategies which can be coded by a genetic individual of length L.

A genetic population is a set of M genetic individuals.[12] In economic models a genetic population represents the whole of the strategies of all economic agents in the same situation at the same time, e.g. the quantity decisions of all suppliers in one period of a cobweb model.

A genetic population consisting of M genetic individuals can be written as a vector

$$\overline{m} = (m(1), m(2), \ldots, m(N)) \tag{3}$$

where $m(i)$ is the absolute frequency of individuals of type i in population $\overline{m}$. Hence,

$$\sum_{i=1}^{N} m(i) = M. \tag{4}$$

Thus, a genetic population can be written and interpreted as a distribution of genetic individuals.

The set of all different genetic populations is S'. There are $|S'| = N'$ different genetic populations:[13]

$$N' = \binom{M+N-1}{M} = \binom{M+N-1}{N-1}. \tag{5}$$

[10]The exact way of decoding the bit string of a genetic individual into its value differs among the models. But — however decoding may work — there is just one fact worth noticing: genetic individuals can only encode a finite number of values, they can never encode the whole continuum of real numbers.

[11]Although the term 'strategy' is widely used in learning related literature, it does not always mean the same as 'strategy' in a game theoretic setting. What is learned is sometimes just a game theoretic 'action'. (For these definitions, see e.g. Rasmussen (1994), pp. 9)

[12]M is often called the size or even the length of a population.

[13]A simple and rather intuitive explanation of the following formula can be found in Nix and Vose (1992), p. 81.

The genetic algorithm itself can be described as a stochastic process which turns one genetic population into another by using certain stochastic operators, namely selection/reproduction, mutation and crossover. It can be shown that a genetic algorithm satisfies the Markov property.[14] Thus, a genetic algorithm is a Markov process.

To gain an as simple as possible insight into the working of a genetic algorithm without loss of precision, this paper will analyze the algorithm piecewise. First a core algorithm will be viewed, the one operator algorithm. Then, in the two following steps two more genetic operators — or learning schemes — will be introduced until the analysis covers the whole genetic algorithm.[15]

3.2 The One Operator Algorithm — Learning by Imitation

The very heart of a genetic algorithm is the selection–/reproduction operator. Selection assigns a fitness value to each genetic individual within the current genetic population. The fitness of an individual gives information about the performance of the individual according to the problem to be solved. The fitness usually equals a value of a function which is optimized by means of the genetic algorithm. In the economic interpretation the fitness shows how good the encoded strategy of an agent really is. An example is Arifovic's (1994) cobweb model, where the fitness is given by the profit an agent earns according to his quantity decision. In this model the objective function to be optimized (precisely: maximized) by means of the genetic algorithm is the individual profit function.

Reproduction means the process of deriving a new population from an old one. Reproduction is done by 'drawing' (with replacement) genetic individuals out of the pool of the old population. Each individual's chance of being drawn is equal to its relative fitness, i.e. the fitness of the individual relative to the average fitness of its population.

From the economic point of view, the assignment of the fitness to each of the individuals is the crucial part of the learning process. It is the fitness of a strategy that decides on being reproduced or not. This central part of the process is often said to be played by the market. The market brings together all agents' behavioral

[14] The Markov property is often called the no–memory–property. A Markov process has no memory in the sense that the probability of reaching one state from another depends only on the current state but in no way on any other state in the history of the stochastic process. More elaborate analysis and explanation can be found in many standard textbooks, e.g. Goodman (1988) or Isaacson and Madsen (1976).

[15] For readers familiar with Dawid's (1996a) analysis a remark seems worthwhile: Dawid's proposition, stating that economic genetic algorithms are special because of their state dependent fitness function, does not have any consequences to the further research in this paper. The described Markov chain will not alter its basic properties, which are time homogeneity and irreducibility.

strategies, evaluates them, and reveals each strategy's quality relative to the present economic environment (which consists of the rest of the population's strategies and the economic problem to be solved). Thus, the role of the market within genetic algorithm learning is mainly the well–known role of being the source of up to then unknown information. The market produces a feedback to every single economic agent which induces him to change his behavior (because of its poor performance) or to keep it (because of its relative success).

Selection and reproduction can be interpreted as a form of learning by imitation: Agents whose strategies lead to relatively poor performance (low relative fitness) give up their former strategy and copy the strategy of a more successful member of the population.

The following Markov chain analysis shows that this simple learning mechanism leads to stable though not always optimal behavior.

The chance of a genetic individual i to be reproduced into next generation's population depends on his relative fitness $P_1\,(i|\overline{n})$, which is the relation of its own fitness to the fitness of the whole population $\overline{n}$. $R(\cdot)$ is the objective function (e.g. the profit function in Arifovic (1994)):

$$P_1\,(i|\overline{n}) = \frac{n(i)R(i)}{\sum_{j\in S} n(j)R(j)}. \tag{6}$$

Consequently, the probability of population $\overline{m}$ to become the direct successor of population $\overline{n}$ by reproduction and selection is

$$P_1\,(\overline{m}|\overline{n}) = \binom{M}{\overline{m}} \prod_{i\in S} P_1\,(i|\overline{n})^{m(i)}, \tag{7}$$

where

$$\binom{M}{\overline{m}} = \frac{M!}{\prod_{i\in S}(m(i)!)}. \tag{8}$$

$P_1\,(\overline{m}|\overline{n})$ is a transition probability of a transition matrix describing the one operator algorithm as a Markov process.

The Markov process characterizing the one operator algorithm has a number of absorbing states. Every uniform population, i.e. every population consisting of only one type of individuals, is an absorbing state. As there are 2^L different genetic individuals there are at least 2^L absorbing states. Every absorbing state can be reached from at least one of the remaining transient states, so that the Markov process will inevitably end up in one of the absorbing states. This means that a one operator genetic algorithm will always lead to a uniform genetic population.

If the one operator algorithm is interpreted as a process of learning by imitation, then this process will evidently lead to a situation where all economic agents follow the same strategy. The result will be an asymptotically stable, uniform pattern of social behavior.

The drawback of this result is the following: Even if the objective function $R(\cdot)$ is totally flat, which means that $\min R(\cdot) = \max R(\cdot)$, the process nevertheless will converge to an absorbing state. In population genetics such a situation is called a situation without selective pressure. From this branch of science it has long been known that the absence of genetic pressure will nevertheless lead to the described phenomenon which is called genetic drift.[16]

Thus, learning by pure imitation will lead to stability, but it can only by pure chance lead to a result which is an optimal solution of the given economic problem.

3.3 The Two Operator Algorithm — Learning by Imitation and Communication

The one operator genetic algorithm can be augmented by a second genetic operator: recombination. The specific recombination operator used here is the one point crossover.[17]

Crossover randomly chooses two genetic individuals ('parents') from their population. It then creates an offspring genetic individual by combining parts of the bit strings of the two parents. In order to do so, the crossover operator randomly cuts the parents' bit strings at a 'cross over point' s, and fits together the first part of the first parent's and second part of the second parent's bit string in order to create the offspring. (Figure 1 shows an example of crossing over two 16–bit parents at a crossover point of $s = 4$.)

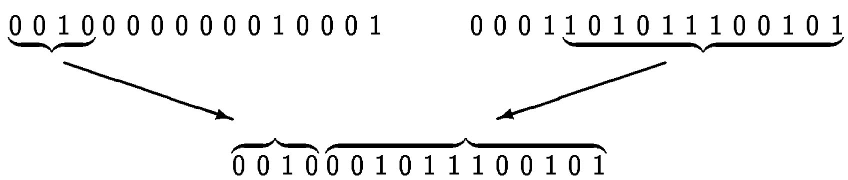

Figure 1: One Point Crossover (example)

Crossover has often been interpreted as a form of learning by communication. Two economic agents meet, talk to each other about their strategies and thus give

[16] See Maynard Smith (1989).

[17] There is in fact a large number of recombination operators (see e.g. Davis (1991)), which can be used within a genetic algorithm. The one point crossover, however, is the recombination operator used in the standard genetic algorithm as well as in most of the other economic genetic algorithm models.

rise to the opportunity of adapting parts of each other's behavior. Although this interpretation seems a bit strange when applied to Arifovic's (1994) cobweb model, there are some models in which learning by communication appears to be sensible. A good example for this is Axelrod's (1987) work concerning evolutionary game theory. In his model, Axelrod uses the genetic individual's bit string to encode a long term 'defect' and 'cooperate' strategy for the repeated prisoner's dilemma.

In technical terms, crossover is a mathematical function $I(\cdot)$. It has four arguments i, j, k and s and returns the value 0 or 1:

$$I(i, j, k, s) \in \{0; 1\}; \quad i, j, k \in S; \quad s \in [1, \ldots, L-1] \tag{9}$$

i and j are the genetic individuals that have be chosen for crossover, i.e. the 'parents' of individual k which is the offspring produced by crossover. s is the crossover point, i.e. the place at which the parents' bit strings are cut. $I(\cdot)$ returns 1, if the offspring k consists of the first part of i and the second part of j. $I(\cdot)$ returns 0, if k consists of the first part of j and the second part of i.[18]

There is a probability χ denoting the chance of an individual getting involved into crossover. Moreover, the crossing point s is set randomly. The possible points $s \in [1, 2, \ldots, L-1]$ are i.i.d.

Thus, probability $P_2(k|\overline{n})$ of obtaining an individual k from population $\overline{n}$ by crossover and selection/reproduction is

$$P_2(k|\overline{n}) = (1-\chi)P_1(k|\overline{n}) + \chi \sum_{i \in S} \sum_{j \in S} P_1(i|\overline{n}) P_1(j|\overline{n}) \frac{1}{L-1} \sum_{s=1}^{L-1} I(i, j, k, s). \tag{10}$$

In analogy to (7), the probability of gaining population $\overline{m}$ directly from $\overline{n}$ is

$$P_2(\overline{m}|\overline{n}) = \binom{M}{\overline{m}} \prod_{i \in S} P_2(i|\overline{n})^{m(i)}. \tag{11}$$

The stochastic process of the crossover–selection/reproduction algorithm is very similar to the one operator algorithm (see 3.2), especially in the impossibility of leaving a uniform population. Again, every uniform genetic population is an absorbing state of the Markov process. The genetic algorithm will inevitably end up in one of these states. Uniform behavior is a situation which cannot be left by means of imitation and communication any more.

[18] In figure 1, if the upper left individual was i and the upper right on was j, the operation could be described as $I(i, j, k, 4)$. If the lower individual was k, the result of the operation shown in figure 1 would be $I(i, j, k, 4) = 1$.

3.4 The Three Operator Algorithm — Learning by Imitation, Communication and Experiments

Finally, a third operator is added to create a three operator algorithm, which is identical to the standard genetic algorithm (Goldberg (1989)). The third operator is mutation. Mutation randomly alters single bits of the bit string by which a genetic individual is coded. As a binary bit can only be changed into its inverse, the changing of a bit is often referred to as 'flipping' the bit. See figure 2 for an example.

```
0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1
               ↓
0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 1
```

Figure 2: Mutation (example)

From an economic perspective, mutation can be viewed as experimentation.[19] The strategy of an economic agent can be slightly changed by altering parts of it. Mutation can support the discovery of totally new strategies. Whereas imitation (selection/reproduction) and communication (crossover) can only reproduce strategies which are already in use (at least partially) by other individuals, experimentation (mutation) is able to find strategies that have never been used before. Mutation is an operator capable of describing true innovation.

The main influence on mutation is the mutation probability μ. μ is the probability of each single bit of an individual's bit string to be flipped.[20]

Moreover, the probability of an individual i to be turned into individual j by mutation only depends on the number of bits that have to be flipped in order to turn i to j. This number is called the Hamming distance between i and j, $H(i,j)$.[21] $H(\cdot)$ is a distance, so that

$$0 \leq H(i,j) \leq L. \tag{12}$$

The probability of turning individual i into j is given by

$$\mu^{H(i,j)}(1-\mu)^{(L-H(i,j))}. \tag{13}$$

[19] An alternative interpretation sees mutation as the result of mistakes in imitation. Mutation arises if an economic agent fails to correctly imitate another agent's strategy. See for example Alchian (1950).

[20] Usually, μ is a very small probability, often set to somewhere between $1/100$ and $1/1000$.

[21] The Hamming distance between the individuals in figure 2 is — certainly — 1.

In the three operator case — using selection/reproduction, crossover and mutation — the probability of gaining individual i from population $\overline{n}$ is

$$P_3(i|\overline{n}) = \sum_{j \in S} \mu^{H(i,j)} (1-\mu)^{(L-H(i,j))} P_2(j|\overline{n}). \tag{14}$$

Defining $\alpha := \frac{\mu}{1-\mu}$ leads to

$$P_3(i|\overline{n}) = \frac{1}{(1+\alpha)^L} \sum_{j \in S} \alpha^{H(i,j)} P_2(j|\overline{n}). \tag{15}$$

Finally, the probability of turning population $\overline{n}$ directly into $\overline{m}$ by using the three operator algorithm is

$$P_3(\overline{m}|\overline{n}) = \binom{M}{\overline{m}} \prod_{i \in S} P_3(i|\overline{n})^{m(i)}. \tag{16}$$

Again, (16) is a transition probability of a transition matrix describing the Markov process of the three operator algorithm.

Unlike for the one and two operator algorithms, there are no absorbing states any more. It can be shown that there is a positive lower bound for $P_3(\overline{m}|\overline{n})$ indicating that $P_3(\overline{m}|\overline{n})$ is strictly positive. Every state of the Markov process can be reached from every other (including the state itself) within even one step. This means that the whole Markov chain consists of only one — transient — class.

The three operator genetic algorithm will not converge into some uniform state. Nevertheless it can be shown that the three operator genetic algorithm *will converge* into a situation with a constant probability distribution of all of its states (i.e. all its populations).

An outline of the proof runs as follows: The one step transition matrix P of the three operator algorithm is regular, which (in this case) means that it is irreducible and aperiodic. For regular matrices there exists a stochastic matrix A with

$$A = \lim_{n \to \infty} P^n, \tag{17}$$

which consists of identical row vectors $\overline{q}_\alpha$. $\overline{q}_\alpha$ gives the constant long run distribution of the states of the Markov process. This means that it characterizes a probability distribution which is not changed by an additional step of the process:

$$\overline{q}_\alpha = \overline{q}_\alpha P. \tag{18}$$

According to the definition of eigensystems, $\overline{q}_\alpha$ is a left eigenvector of P to P's eigenvalue of $\lambda = 1$.

The Perron–Frobenius theorem for regular stochastic matrices[22] ensures that there exists one unique (due to multiplication) such vector $\overline{q}_\alpha$. The stationary distribution of states is characterized by that vector $\overline{q}_\alpha$ the elements of which sum up to unity:

$$\sum_{i=1}^{N} \overline{q}_{\alpha,i} = 1 \tag{19}$$

From this it can be seen that there is a unique and constant long–run distribution of *populations* which is reached by the three operator algorithm. The specific form of this distribution is described by the vector $\overline{q}_\alpha$.

This result looks a bit tricky. It is important to keep in mind the two distinct distributions talked about. First, a population is a distribution of genetic individuals. Or — in economic terms — a population is a vector of individual behavioral strategies, i.e. a *social* behavioral pattern. Second, and different from the first, vector $\overline{q}_\alpha$ is a distribution of populations, i.e. of different social behavioral patterns.

In plain mathematical terms, the three–operator–algorithm ends up in a state which is Ljapunov stable: There is a set of populations with a positive long run measure, and this set is a true subset of the set of all populations.[23]

A learning scheme consisting of learning by imitation, communication and experiments displays two basic properties: On the one hand such a learning scheme will never lead to asymptotic stability in behavior. Economic agents using such a scheme will never stop experimenting and consequently will never stop trying new strategies. On the other hand, in the long run all kinds of experiments will occur with a certain constant probability. Thus, after a while, all *combinations* of individual behavioral strategies will occur with a fixed probability. This does not mean that social behavior (or aggregate data characterizing social behavior) will remain unchanged for the rest of times. But it also does not mean that there will be erratic behavior for the rest of times. Ljapunov stability of genetic algorithm learning simply means that long run social behavior will remain within a certain corridor of different social behavioral patterns.

[22]See, for example, Isaacson and Madsen (1976), pp. 123–132.

[23]Ljapunov stability in the space of populations can also be interpreted as asymptotic stability in the space of long run *distributions of populations*: In the long run, a constant and unchanging distribution of populations is reached.

4 Genetic Algorithm Learning under the Regime of the Market

4.1 Some Basic Notions

While the previous section contains a rather formal analysis, this section switches to an interpretation of genetic algorithms as learning processes. Specifically, GA learning will be interpreted when taking place under the regime of markets. In this section, a few concepts of evolutionary game theory will be used. Nevertheless, the relation of these concepts to genetic algorithm learning will only be briefly characterized. For a more detailed description of the similarities of GA learning and evolutionary game theory, see Birchenhall, Kastrinos, and Metcalfe (1997) or Riechmann (1998).

First of all, it is important to recognize the corresponding application of the population concept in genetic algorithm learning as well as in evolutionary game theory. In game theory, a population describes a set of players, each playing a pure strategy in some (economic) game. An even shorter description says that a population is a frequency distribution of pure strategies played within a society. Note that both interpretations are exactly the same as the ones used for genetic populations in the preceding section (especially equation (3)).

Further analysis of GA learning with respect to game theory shows that even more can be said. Every economic agent tries to maximize his performance relative to his environment, which can entirely be described by his objective $R(\cdot)$ (see equation (6)) and the rest of his population $\overline{n}$. Thus, every economic agent faces the problem

$$\max_{i \in Z} R(i|\overline{n}); \quad Z \subseteq S, \tag{20}$$

where Z is the set of all economic strategies available to the agent. Z is a subset of S, the set of all possible strategies. Equation (20) is the definition of a Nash strategy: It claims the agent to chose strategy i of all strategies Z available to him which represents the best response to all the other agents' strategies. Thus, every agent tries to play Nash, and selection (which is essentially driven by the market) works in favour of Nash strategies. But whether or not a Nash equilibrium or even a series of moving Nash equilibria can be obtained by GA learning depends on Z, which is the set of strategies available to the agents. As will be pointed out in greater detail a bit later, Z in turn depends on the learning mechanisms the agents are allowed to learn. Markov chain analysis shows that only learning by experiments is capable of maintaining a strategy set Z that is always the same as S.

Evolutionary game theory offers another tool which is valuable for a more intuitive notion of GA learning as a dynamic process: the notion of evolutionary stability. In short, a population is of evolutionarily stable composition, if it recovers from the infection of an invading strategy within finite time. Although not directly applicable to genetic algorithms, the concept evolutionary stability can be transferred to GA learning without losing its basic idea.[24] Using the idea of evolutionary stability enables focusing the dynamics of genetic algorithm learning from a different point of view: Genetic populations are continuously challenged by one or more invading strategies. These strategies either stay within the population, by that improving its composition[25], or they are rejected again. This process of invasion and possible rejection continues forever. Even if an evolutionarily stable genetic population is reached, invasion will still go on, but now every invader will be rejected. Again it depends on the learning mechanisms that are used if an evolutionarily stable kind of social behavior (i.e. evolutionarily stable genetic population, if one exists) can be gained or not.

It has already been shown in the preceding sections that genetic algorithm learning is a compound learning strategy consisting of three different types of learning. Each of the three types has its own effect on the result of the learning process. This touches two aspects, the stability of the learning process and the quality of the behavior which will been learned.

As the major aim of this paper is to examine the dynamics and stability properties of genetic algorithm learning, the second aspect (quality of learned behavior) will be described rather intuitively. For a more drastical, i.e. mathematical description, refer to Davis and Principe (1993), Nix and Vose (1992), Rudolph (1994), Goldberg and Segrest (1987) and Dawid (1996a).

4.2 Learning by Imitation

As pointed out in section 3.2, learning by imitation leads to behavioral stability in its strongest form, i.e. uniform behavior of all agents. There is simply no chance of learning other strategies than those which were already contained within the first genetic population. (You can only imitate what is already there — that simply is the true meaning of ‘imitation’.) Moreover, the phenomenon of genetic drift will inevidently lead to asymptotically stable, homogeneous behavior.[26]

This also implies that it is impossible to find a better strategy than the best one contained within the very first genetic population. Learning by imitation does lead to a stable outcome of the learning process but it does not necessarily lead to an

[24] For details of evolutionary stability and genetic algorithms refer to Riechmann (1998).

[25] A population is even said to be invaded, if there is nothing more than a change in frequency of the strategies contained within it, which means that invasion does not necessarily mean the occurence of new strategies.

[26] See Björnerstedt and Weibull (1996).

optimal one.

The same applies to the notions of evolutionary dynamics: Only those Nash equilibria can be learned (by society) which consist of strategies already contained within the first population. More precisely: The set Z, which consists of all strategies learnable by imitation, is the set of strategies contained within the first population. Only strategies contained within Z can be learned.

The final population the learning process converges to is evolutionarily stable only due the restricted set of learning mechanisms that economic agents are allowed to use. There may be populations which are evolutionarily superior to the one which is finally learned. But these superior populations cannot be reached by mere imitation, because they contain strategies which are not contained within Z.

In order to clarify the numerical dimensions talked about, imagine a genetic population consisting of M genetic individuals of length L. Such a population can at best consist of M different genetic individuals, i.e. represent no more than M different behavioral strategies ($|Z| \leq M$.) If only learning by imitation takes places, only the best of these M strategies can be found, which is a rather poor result compared to the fact that in this case a number of $|S| = 2^L$ (see equation (2)) different behavioral strategies *can* exist.[27]

4.3 Learning by Communication

Learning by communication is a process very similar to learning by imitation.[28] Specifically, its outcome is predetermined by the first genetic population which determines Z, the set of strategies learnable by communication. Although it is possible to exchange parts of the strategies (i.e. parts of the bit strings), only those parts can be exchanged which were already contained in the first genetic population (or in Z). If nobody knows about a certain detail which could make a good strategy a perfect one, then nobody can acquire this detail by communication.[29]

Thus, learning by communication is able to find better strategies than learning by imitation, because the best details of different strategies can be combined. But still it is impossible to find a strategy parts of which were not contained within the first population.

[27] As an illustration: Taking a genetic population of $M = 100$ individuals of length $L = 16$, only about 0.0015% of all strategies can be learned by pure imitation.

[28] In fact, learning by communication is nothing more than ‘imitation of parts’ or, as Birchenhall (1995) puts it, ‘modular imitation’.

[29] The same argument in its technical form reads: If, for instance, the bit string of the perfect strategy requires an ‘0’ in its second place, but every first population genetic individual’s bit string contains a ‘1’ in second place, then there is no way of finding the perfect strategy by just using the crossover operator.

As a second aspect, a uniform population still cannot be altered by communication. If all agents behave the same, there are no different details of strategies to be exchanged.

In the end of the learning process, learning by communication leads to a result which is homogeneous and asymptotically stable, and which in most cases leads to better results than learning by imitation, but which still is highly dependent on the behavioral patterns at the very start of the process. Again, not every Nash equilibrium can be learned and the final population may only be evolutionarily stable with regards to the learning mechanisms available to the agents and by that with regards to Z, which may be smaller than the set S of all possible behavioral strategies.

In some more technical papers (i.e. Goldberg (1989), Holland (1996)), crossover or learning by communication is seen to be the main force to accelerate learning by exploring the search space.[30] It is pointed out that learning of relatively good behavioral strategies is much faster with communication involved compared to learning by experiments alone. Additionally, remembering the importance of heterogeneity for learning by communication, it is easy to conclude that there is a close connection between the extend of heterogeneity within a population and the time it takes to find better behavioral strategies: The more diverse a population is, the faster can a good social behavior be learned.[31]

Again imagining the dimensions, learning by communication alone searches up to $|Z| \leq 2\binom{M}{2}(L-1)$ alternative strategies, which can result from crossover based on the very first population.[32]

4.4 Learning by Experimentation

Learning by experimentation is quite different from learning by communication and learning by imitation. Experiments can find patterns of behavior that have never occurred in society (i.e. genetic population) before. Experimentation allows for the development of strategies which have not been used — not even partially — by any earlier population. Experimentation is capable of finding true innovation.[33] This makes it possible to win better strategies than by imitation or communication. Genetic algorithm research shows that the standard genetic algorithm is able to find

[30] See Romer (1992) for a description of learning as searching a very large search space.

[31] This may even be a behavioral interpretation of the Fisher principle (Birchenhall et al. (1997)), which links the rate of fitness growth within a population to the population's variance (see e.g. Maynard Smith (1989)).

[32] This means that from about $M > 10$ a population is almost surely large enough to contain all the information needed to learn by communication every possible strategy. Still the fact that a population is large enough is no guarantee that it really does contain all the information required.

[33] For a similar point of view, see Blume and Easley (1993).

optimal strategies for a large number of problems. And, more than this, these optimal strategies are found irrespectively of the starting population. The irrelevance of the starting conditions is one of the central features of the mutation operator.[34]

Markov chain analysis shows that learning by experiments is a process that will not fully converge to a uniform population. Totally homogeneous behavior or at least an unchanging population will never be established. Instead, there will always be individual experimentation which disturbs the composition of the population. At least, the disturbance will happen in a regular manner, gaining a final distribution of social behavior (i.e. a distribution of genetic populations) which is constant and independent of the starting conditions.

Markov chain analysis reveals the ambiguity of the mutation operator: On the one hand, it supports finding better behavioral patterns, but on the other hand, it prevents social behavior from complete convergence. In technical terms, the resulting stability is of a Ljapunov stable type, which means that in the long run only a subset of all possible social behavioral patterns will be adopted by society.

More detailed technical research shows that even learning by experimentation results in *asymptotically stable* behavior, if either the number of economic agents is infinitely large (Nix and Vose (1992)) or if the frequency of experiments declines in time (Davis and Principe (1993)).

Moreover, learning by experimentation is the only learning technique which is able to reach all strategies, which means that the set of strategies learnable by experimentation is the same as the set of all strategies: $Z = S$. Thus, only learning by experimentation can lead to every possible Nash equilibrium as it will lead to an evolutionarily stable population, provided there exists one.

4.5 The Compound: Genetic Algorithm Learning

Thus, does genetic algorithm learning lead to behavioral stability?

Genetic algorithm learning does even lead to asymptotic behavioral stability — as long as there is no experimentation or experimentation ceases in time. If this is the case, learning dynamics is the one presented for the two operator case (see

[34] This may be regarded as a weakness of the concept of genetic algorithm learning, as it neglects the possibility of modelling path dependence or lock–ins. So it may be worthwhile to mention two further points, which are mainly beyond the scope of this paper. First, depending on the underlying (economic) problem, some GAs spend long times supporting populations which are not evolutionarily stable. Some keywords pointing to this topic are ‘deceptiveness’ of genetic algorithms and the problem of ‘premature convergence’. Secondly, the lack of ability to model lasting lock–ins or path dependence applies to the basic genetic algorithm. There are variations of genetic algorithms which are capable of modelling these phenomena. One keyword pointing into this direction of research may be ‘niching mechanisms’. Again, a good starting point for more descriptions of all of the special cases and variants of GAs is Goldberg (1989).

section 3.3). The long run result of genetic algorithm learning without mutation or with declining mutation rate is a state owning two important properties: Every economic agent behaves just like every other one does, and every economic agent behaves just like he did last period. Behavior will be the same over all individuals and for all remaining periods.

As soon as experimentation is involved, the picture changes dramatically: The only thing that can be said is that different social patterns of behavior (i.e. genetic populations) have a constant long term probability distribution. This means that the chance of each social behavioral pattern to show up again next period will be the same every period until the end of times. This property is far away from being stable in the true sense of the concept of asymptotic stability, although it is also far away from being unstable in the sense that there is no fixed long run probability distribution of the different patterns of social behavior.

Thus, from Markov chain analysis it can be concluded that genetic algorithm learning simply ends in a state which is Ljapunov stable. Further investigations help explaining the reasons for this.

With all three types of learning involved, the market as the coordinator of the agents' strategies gains a crucial role for stability of social learning schemes. As long as no mutation or learning by experiments takes place, no selective force (i.e. no market) is needed: Because of genetic drift, convergence of behavioral strategies will be established completely without it.[35] As soon as experimentation enters the stage, a force is needed, that can serve as a counterpart to experimentation's continuous disturbance of social behavior. It is the informational function of the market combined with the selective pressure caused by it that plays this role.[36] Every time experimentation introduces a behavioral strategy which leads to a worsening of social behavior (leading to an evolutionarily worse population), the market makes sure that this strategy will disappear again.

Thus, taking all three learning techniques together and letting them act within an environment of markets, two opposite forces can be identified to be the source of genetic algorithm's characteristic long run dynamics: Learning by experimentation, which continuously disturbs behavioral stability, and coordination of the market, which continuously gains stability back again.

[35]This of course does not mean that without selection or market the best alternative is learned.

[36]This notion is similar to Hayek's (1978) view of the role of the market within the process of economic competition.

5 Summary

If genetic algorithms are to be used as a metaphor for economic learning, there are some things to be aware of. The main thing is the fact that genetic learning is a compound of three types of learning techniques, each of which behaves differently in its own. It has been shown that it is the combination of all the three that — in presence of the market — gives the standard genetic algorithm its characteristic dynamic properties.

Learning according to a genetic algorithm will not lead to a situation in which each economic agent behaves the same or at least the social behavioral pattern remains unchanged. Though most agents will show up a similar behavior there will always be some agents which change their behavior by experimenting. Thus, on the one hand there is learning by imitation and communication that forces individual behavior to converge. On the other hand there is learning by experiments that leads some individuals' strategies to to be altered and thus diverge from most of the others' behavior. Above all, there is the market, continuously controlling and coordinating agents' behavior, thus allowing change if it is advantageous and rejection it if it is not.

Thus, genetic algorithm learning in fact offers a way of modelling what Witt (1993, p. xix) calls an *'interplay of coordinating tendencies arising from competitive adaptions in the market and de–coordinating tendencies caused by the introduction of novelty.'*

References

Alchian, Armen A. (1950). Uncertainty, Evolution, and Economic Theory. *Journal of Political Economy*, **58**, 211–221.

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

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

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

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

Axelrod, Robert (1987). The Evolution of Strategies in the Iterated Prisoner's Dilemma. In: *Genetic Algorithms and Simulated Annealing*, edited by Davis, Lawrence, pp. 32–41. Pitman, London.

Azariadis, Costas (1993). *Intertemporal Macroeconomics*. Blackwell, Oxford, UK.

Birchenhall, Chris (1995). Modular Technical Change and Genetic Algorithms. *Computational Economics*, **8**, 233–253.

Birchenhall, Chris, Kastrinos, Nikos, and Metcalfe, Stan (1997). Genetic Algorithms in Evolutionary Modelling. *Journal of Evolutionary Economics*, **7**, 375–393.

Björnerstedt, Jonas and Weibull, Jörgen W. (1996). Nash Equilibrium and Evolution by Imitation. In: *The Rational Foundations of Economic Behaviour*, edited by Arrow, Kenneth J., Colombatto, Enrico, Perlman, Mark, and Schmidt, Christian, chap. 7, pp. 155–171. MacMillan, London.

Blume, Lawrence E. and Easley, David (1993). Economic Natural Selection. *Economics Letters*, **42**, 281–289.

Bullard, James and Duffy, John (1998). A Model of Learning and Emulation with Artificial Adaptive Agents. *Journal of Economic Dynamics and Control*, **22**, 179–207.

Clemens, Christiane and Riechmann, Thomas (1996). Evolutionäre Optimierungsverfahren und ihr Einsatz in der ökonomischen Forschung. Diskussionspapier 195, Universität Hannover, Fachbereich Wirtschaftswissenschaften.

Davis, Lawrence (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York.

Davis, Thomas E. and Principe, Jose C. (1993). A Markov Chain Framework for the Simple Genetic Algorithm. *Evolutionary Computation*, **1**, 269–288.

Dawid, Herbert (1996a). *Adaptive Learning by Genetic Algorithms*. Springer, Berlin, Heidelberg, New York.

Dawid, Herbert (1996b). Genetic Algorithms as a Model of Adaptive Learning in Economic Systems. *Central European Journal for Operations Research and Economics*, **4**, 7–23.

Franke, Reiner (1997). Behavioural Heterogeneity and Genetic Algorithm Learning in the Cobweb Model. Discussion Paper 9, IKSF — Fachbereich 7 — Wirtschaftswissenschaft. Universität Bremen.

Goldberg, David E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison–Wesley, Reading, Massachusetts.

Goldberg, David E. and Segrest, Philip (1987). Finite Markov Chain Analysis of Genetic Algorithms. In: *Proceedings of the Second International Conference on Genetic Algorithms*, edited by Grefenstette, John J., pp. 1–13.

Goodman, Roe (1988). *Introduction to Stochastic Models*. Benjamin/Cummings, Menloe Park, California.

Hayek, Friedrich A. von (1978). *New Studies in Philosophy, Politics, Economics and the History of Ideas*, chap. 12, Competition as a Discovery Process, pp. 179–190. Routledge & Kegan Paul, London.

Heinemann, Maik (1998). Adaptive Learning of Rational Expectations Using Neural Networks. forthcoming, Journal of Economic Dynamics and Control.

Hofbauer, Josef and Sigmund, Karl (1988). *The Theory of Evolution and Dynamical Systems*. Cambridge University Press, Cambridge, UK.

Holland, John H. (1996). The Rationalityof Adaptive Agents. In: *The Rational Foundations of Economic Behaviour*, edited by Arrow, Kenneth J., Colombatto, Enrico, Perlman, Mark, and Schmidt, Christian, chap. 12, pp. 281–297. MacMillan, London.

Isaacson, Dean L. and Madsen, Richard W. (1976). *Markov Chains, Theory and Applications*. John Wiley & Sons, New York.

Lucas, Robert E. Jr. (1986). Adaptive Behavior and Economic Theory. *Journal of Business*, **59**, S401–S426.

Marcet, Albert and Sargent, Thomas J. (1989). Least–Squares Learning and the Dynamics of Hyperinflation. In: *Economic Complexity: Chaos, Sunspots, Bubbles, and Nonlinearity*, edited by Barnett, William A., Geweke, J., and Shell, K., chap. 7, pp. 119–137. Cambridge University Press, Cambridge, UK.

Maynard Smith, John (1982). *Evolution and the Theory of Games*. Cambridge University Press, Cambridge, UK.

Maynard Smith, John (1989). *Evolutionary Genetics*. Oxford University Press, Oxford, UK.

Michell, Melanie (1996). *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, Mass., London, England.

Nix, Allen E. and Vose, Michael D. (1992). Modeling Genetic Algorithms with Markov Chains. *Annals of Mathematics and Artificial Intelligence*, **5**, 79–88.

Rasmussen, Eric (1994). *Games and Information. An Introduction to Game Theory*. Blackwell, Oxford, UK, 2nd edn.

Riechmann, Thomas (1997). A Genetic Algorithm Model of Natural Monopoly. mimeo.

Riechmann, Thomas (1998). Genetic Algorithm Learning as an Evolutionary Process. Diskussionspapier, Universität Hannover, Fachbereich Wirtschaftswissenschaften.

Romer, Paul M. (1992). Two Strategies for Economic Development: Using Ideas and Producing Ideas. In: *Proceedings of the World Bank Annual Conference on Development Economics 1992*, pp. 63–91.

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

Sargent, Thomas J. (1993). *Bounded Rationality in Macroeconomics*. Claredon Press, Oxford.

Weibull, Jörgen W. (1995). *Evolutionary Game Theory*. MIT Press, Cambridge, Mass., London, England.

Witt, Ulrich (1993). Introduction. In: *Evolutionary Economics*, edited by Witt, Ulrich, pp. xiii–xxvii. Edward Elgar, Aldershot, England.