A Genetic Algorithm for Windrum-Birchenhall Evolutionary Economic Model

Assistant lecturer, PhD. Costin-Radu BOLDEA
Informatics Department, University of the West of Timisoara

A genetic evolutionary algorithm for a modified version of the Windrum-Birchenhall neo-Schumpeterian model of industrial dynamics is proposed.

Keywords: *evolutionary algorithms, evolution strategies, agent-based computational economics.*

1 Introduction

The last years have been seen a huge flourishing of works studying evolutionary methods behavior in diverse fields. Following the computer innovations, there has been a growing interest in application to economic models of learning procedure developed in evolutionary computation tools such as genetic algorithms. Accordingly then, the use of computer simulation based on the related genetic algorithms (GAs) has largely taken by many researchers, for example, Axelord (1987 [4]), Arifovic (1994, 1995[1, 2, 3]), Dawid (1994, 1996 [6, 7, 8]), and Birchenhall (1994 [5]).

Such works may illustrate an uneasy acceptance of the assumption of perfect foresight or rational expectation. Under the assumption, the analysis of the single representative agent in economic modeling may produce an inconsistency with interpretations of results of general equilibrium analysis. However, while the perfect foresight and rational expectation assumptions have became a standard feature of general equilibrium economic theory, the equilibrium that are optimal and determinate will fail in an overlapping generation economy.

We study a simple economic model as an adaptive learning system. There are two populations co-existing in each period of time. A significant departure to representative agent in economic modeling is a relaxation of hypothesis of rational expectations. Individual agents have heterogeneous beliefs concerning realization of possible outcomes. With the existence of heterogeneity in the economy, the actual outcome may or may not be identical to any particular individual agent's expectation. When the actual outcome feeds back to individual agents' beliefs, individual agents learn to adaptively adjust their own beliefs. The learning is via a so-called *genetic algorithm process*.

2. Basics of Evolutionary Genetic Algorithms

The Genetic Algorithm (GA) is a computational model of evolution, currently the most prominent and widely used model of evolution in artificial-life systems. The GA uses Darwin's basic principles of natural selection and mutation, and a cross breeding to create solutions for problems, in general Here we consider the GA as an economically and socially meaningful model of adaptive learning. Technically speaking, the GA is a search algorithm and complementary tool for optimizing problems. The GA functioned as a highly parallel mathematical algorithm that transformed a population of individual mathematical entities, each with an associated fitness value, into a new population. The GA operates after the Darwinian principles of natural selection and *"survival of fittest*", and after naturally occurring genetic operations.

The evolutionary process of the GA has been adequately used to model the adaptive behavior of a population of bounded rational agents interacting within an economic system.

There comes to a connection the theory of genetic algorithm leaning to evolutionary game theory. The basic argument is in the discussion of property of stability of genetic algorithms. Riechmann (1998 [8]) argues that a concept of evolutionary stability will be:

*"A population is evolutionarily stable if it is resistant against changes in its compositio*n." Selection alone cannot make exchanging of concepts in the process of learning. What is appealing intuitively in the GA is that crossover and mutation combine to search potentially pregnant new concepts. Dawid (1996) gave an excellent interpretation on these. The whole process in the genetic algorithm makes up the building-block structure in which all agents' beliefs are updated.

Nowadays, there are many variations of the genetic algorithms. However, most of these variations still keep the original principles of Holland's GA. Three main genetic operators, selection, recombination and mutation, constitute such a framework of the genetic algorithm learning as the standard genetic algorithm (SGA),

3. The Economic Evolutionary Model of Windrum-Birchenhall

A two-population model that comprises a population of firms and a population of consumers, which adapt to and learn about preferences of each other, was presented by Paul Windrum and Chris Birchenhall (Windrum, Birchenhall [11]). The adaptive learning is mediated by the technological designs that are traded in the market. A unit of evolution is a firm and a unit of selection is a product design.

Selection process acts through adaptive learning of both populations. The main feature of the model is that it simulates interaction of a number of consumers (partitioned into a number of types) and producers (firms). In each period the following sequence is repeated:

- Consumers allocate their purchases across the firms (out of firms' offer of sale which consist of maximum quantity of goods and their prices);
- Population of firm replicates and adjusts their offer;
- Consumers make adjustment of their attempting.

Each consumer will attempt to buy the most attractive offer if this offer is better than the option named 'not buying' (if there is no stock available a consumer tries to buy the second best, etc). A consumer of type *i* is characterized by a quantity of money m_i and a utility function of the form:

$$u_i(x,p)=v_i(m_i-p)+w_i(x)$$

where x is the characteristic vector of a good and p is the price of the good; v is the indirect utility of money that can be obtained in other markets (this function has a constant form); w(x) is the direct utility of consuming the good with characteristic vector x.

Distribution of consumers across the set of types is governed by a form of equation in such a way that proportion of the consumers expectations (share) on the type *i products* grows in proportion to the utility attained by that type (related to the maximum value of the utility at current time t).

After trading, the firms adjust their offer by modifying prices and quantities produced. Price markup and target market shares are fixed in the model so the only task is to adjust the firm's capacity and level of production. Investment of a firm depends on the wealth of the firm, which is defined as all collected profits (or losses) of that firm during its life.

Firms can innovate, i.e. can modify products characteristics *x* (each characteristics can mutate with given probability, and if it mutates, characteristics values are randomly changed accordingly to assumed normal distribution). The firm compares the mutated design with the old design and allows the mutated design to be put into production if it is assessed to be improvement. A kind of imitation of existing designs is present in the model; it takes a form of transfer of values of selected number of characteristics from randomly selected firm to that being "under imitation procedure". The transfer is accepted if it improves the utility of target consumer type. We can say that the innovation type present in the model is a product innovation embodied in its design.

The model explains the influence of the consumer preferences (ignored in the traditional product life cycle).

4. Proposed Genetic Algorithm

As in natural ecosystems, GAs typically evolve a population of individuals. Here, each individual is a bargaining strategy of the adaptive agent. Our evolutionary model consists of two co-evolving agents (where each agent maintains its own collection of strategies). We assume that one of the agents, denoted as “agent 1” or the "firm-agent", has the privilege to open the negotiations. In reality this situation frequently occurs when a potential client wants to buy something from a professional seller. Normally, the seller takes the initiative: he or she can either refer to the indicated price on the product, or propose an initial price.

Like in nature, the survival probability of each bargaining strategy depends on its fitness (the “survival of the fittest” concept). During the fitness evaluation, each strategy competes against a group of opponent strategies who are drawn at random (without replacement) from the population of strategies of the other agent. The fitness is then equal to the mean payoff obtained against these opponent strategies.

Genetic representation

In our model, each firm-agent specifies a list of offers and thresholds for the different selling rounds. The thresholds determine whether an offer of the other party is accepted or rejected. Each firm-agent is encoded as a sequence of real-coded genes (together called a “chromosome”) in our evolutionary system. This representation is following

$$\text{Firm\_Agent}(j) \rightarrow c_1(j)^{(2\text{bytes})}p_1(j)^{(2\text{bytes})})\ldots c_n(j)\,p_n(j)$$

where c_i is the quantity in stock of the *i* good, p_i is his price, used here like characteristic value of the product, and *n* is the number of goods available (fixed).

For the consumer-agent, we use an representation of the form

$$\text{Consumer}(k) \rightarrow m^{(4\text{bytes}})q_1(k)^{(2\text{bytes})}\ldots q_n(k)^{(2\text{bytes})}$$

where *m* is his quantity of money, and q is the percent of money allowed to buy the *i* – good, i. e. the relative utility value of the product

$$q_i = \frac{u_i}{\sum_{ki} u_k}.$$

In first step, the algorithm calculates the *fitness* for-each firm agent, using a *negotiation round*: all the consumers allocate their purchases across the firms. Each consumer will attempt to buy the most attractive offer for a product (if there is no enough stock available a consumer tries to buy the second best). For each firm-agent, the sum of all offers for a product *i* is the *utility functions* for this product, and the *fitness function* will be f=å s_i.

For the consumers, the algorithm selects and modifies the utility function of products:

$$u_i(k)=m(k)\,(1-q_i)+w_i(k)$$

where w_i is the rapport between the sum of all demands and the sum of values of the quantities available:

$$w_i = \frac{\sum_k m(k)q_i(k)}{\sum_j c_i(j)p_i(j)}$$

The quantity of money *m(k)* for a consumer is randomly generated for each round, using a Gauss distribution around a fixed average value *m*.

Selection operator

Selection is performed using the (μ+? ?selection scheme over the population of firm-agents. In conventional notation, μ is the number of parents and ? ?is the number of generated offspring *(*μ = ? ? ? ?for example). The μ survivors with the highest fitness are selected from the initial population P_{old}. For this ones, an adjustment of their chromosomes is operated by modification of the capacity of production (the level of each stock), depending on the firm’s profits (and losses). The stock will be proportionally increased with the rapport between the value of the demand of the product and the average value of all demands, multiplied by the number of goods.

The rest of the new agents’ population P_{new} will be produced by mutation operation.

An offspring agent is generated in two steps. First, an agent in the rest of the population is (at random, with replacement) selected to be a parent. The chromosome of this parental strategy is then mutated to generate a new offspring agent (the mutation model is specified below).

Mutation operator

After the selection is performed, the mutation takes place. This is to prevent falling all solutions in the population into a local optimum of solved problem. Mutation changes randomly the new offspring. For the described encoding we can modify few values of stocks randomly choused between two extreme values c*min* and *cmax*. Mutation probability says how often will be parts of chromosome mutated, i. e. the relative number of product stocks affected If there is no mutation, offspring is taken after crossover (or copy) without any change. If mutation is performed, part of chromosome is changed. If mutation probability is 100%, whole chromosome is changed, if it is 0%, nothing is changed. Mutation is made to prevent falling GA into local extreme, but it should not occur very often, because then GA will in fact change to random search.

A strategy of mutation can be used for adjusting the prices too. Mutation can be interpreted as undirected exploration of new strategies, or as mistakes made during imitation. It is important to note that, in our model, the agent's strategies are not binary strings (as in most GA implementations) but, instead, consist of strings of real-coded numbers.

5. Simulation Result

Our study focuses on an exhaustive simulation and investigates the performances of these learning algorithms. The entire data set of simulations in this study is available from the author upon request. When our study may fit into a research program of agent-based modeling, the investigation of simulation results is in many ways.

The result of the simulation shows that, in most of the experiments, an expectation (demands) equilibrium of the model emerged. In some experiments, other convergence results are emerged. There are some experiments that convergence fails to obtain within our simulation criterion.

6. Conclusion

In applying the computational algorithm to the adaptive learning system, interpretation is both more and less limited. As the methodological role of computer simulations in studying economic models is not well developed, some researchers give little weight to and question the reliability of such work. The major advantage is that we can study models that do not involve the restrictive assumptions that would be required to produce analytical results.

References

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

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

3. Arifovic, J. and C., Eaton (1995) "Coordination via genetic learning," *Computational Economics* 8, 181-203.

4. Axelrod, R., (1990) *The Evolution of Cooperatio*n, Penguin.

5. Birchenhall, C. R. (1994) "Evolutionary Games and Genetic Algorithms," *School of Economics Discussion Pape*r, University of Manchester.

6. Dawid, H. (1994) "A markov chain analysis of genetic algorithms with a state depenent fitness function," *Complex System* 8, 497-417.

7. Dawid, H., (1996) "Genetic algorithms as a model of adaptive learning in economic systems," *Central European Journal for Operations Research and Economics* 4(1).

8. Dawid, H., (1996) "Learning of cycles and sunspot equilibrium by genetic algorithms", *Journal of Evolutionary Economics* 6, 361-373.

9. Nelson, R.R., and Winter, S.G. (1982), *An Evolutionary Theory of Economic Chang*e, Belknap Press, Cambridge, Mass. and London.

10. Riechmann, Thomas (1998) "Genetic Algorithm and economic Evolution", Diskussionpapier no. 219, ISSN 0949-9962.

11. Windrum Paul, Chris Birchenhall, (1998), 'Is product life cycle theory a special case? Dominant design and the emergence of market niches through coevolutionary-learning, *Structural Change and Economic Dynamic*s, 9, pp. 109-134.