Centrum voor Wiskunde en Informatica

REPORTRAPPORT

SEN

Software Engineering

Software ENgineering

SEN

The separation of economic versus EA parameters in EA-learning

F. Alkemade, J.A. La Poutré, H.M. Amman

REPORT SEN-E0416 SEPTEMBER 2004

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-369X

The separation of economic versus EA parameters in EA-learning

ABSTRACT
Agent-based computational economics (ACE) combines elements from economics and computer science. In this paper, we focus on the relation between the evolutionary technique that is used and the economic problem that is modeled. Current economic simulations often derive parameter settings for the genetic algorithm directly from the values of the economic model parameters. In this paper we show that this practice may hinder the performance of the GA and thereby hinder agent learning. More specifically, we show that economic model parameters and evolutionary algorithm parameters should be treated separately by comparing two widely used approaches to population learning with respect to their convergence properties and robustness.

2000 Mathematics Subject Classification: 91-08
Keywords and Phrases: Genetic Algorithms;Economics;Robustness

The separation of Economic versus EA parameters in EA-learning

Floortje Alkemade[(1)] Han La Poutré[(1,2)] Hans Amman[(2)]

(1) CWI, Centre for Mathematics and Computer Science
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
(2) Eindhoven University of Technology
School of Technology Management
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

Abstract

Agent-based computational economics (ACE) combines elements from economics and computer science. In this paper, we focus on the relation between the evolutionary technique that is used and the economic problem that is modeled. Current economic simulations often derive parameter settings for the genetic algorithm directly from the values of the economic model parameters. In this paper we show that this practice may hinder the performance of the GA and thereby hinder agent learning. More specifically, we show that economic model parameters and evolutionary algorithm parameters should be treated separately by comparing two widely used approaches to population learning with respect to their convergence properties and robustness.
Keywords: Genetic Algorithms, Economics, Robustness

1 Introduction

Agent-based computational economics (ACE) concerns the computational study of economies modeled as evolving systems of autonomous interacting agents [37]. The ACE approach combines elements from evolutionary economics, cognitive science and computer science. Evolution-inspired algorithms, such as genetic algorithms are increasingly used in ACE [2, 3, 4, 5, 19, 20, 11, 6, 30, 31, 15, 16, 14, 33, 38, 1, 21, 28, 12, 44, 10, 36, 17, 35]. Evolutionary algorithms are biologically inspired techniques that use the concept of "survival of the fittest" to evolve agent behaviour that becomes better and better adapted to the environment (i.e., a particular market). Evolutionary algorithms thus provide us with a way to model boundedly rational agent learning and decision making. The popularity of evolutionary algorithms in economic simulations can be attributed to the fact that they allow us to model large systems of boundedly rational agents from the bottom up. Evolutionary algorithms were originally designed for and used in optimization problems and there is a vast literature that describes how to tune the algorithm to perform well in optimization problems [26, 24, 8, 39]. Now that evolutionary algorithms are also used in the social simulation domain these guidelines have to be reestablished. Current economic simulations often derive parameter settings for the genetic algorithm directly from the values of the economic model parameters. In this paper we show that this practice may hinder the performance of the GA and thereby hinder agent learning. Furthermore, we show that in order to obtain robust results economic model parameters and evolutionary algorithm parameters should be treated separately by comparing two widely used approaches to evolutionary algorithm learning.

In the first approach the evolutionary population is considered as a population of agents (one chromosome equals one agent) whereas in the second approach the evolutionary population is considered as a population of strategies from which the agents can choose. We show that the first approach (direct evolutionary interpretation of the economic model) may lead to premature convergence of the genetic algorithm. Robustness to variations in the parameter settings is particularly important in social simulation since it is often difficult to give a direct economic interpretation of a particular evolutionary algorithm parameter (such as population size or recombination rate). This makes it impossible to interpret the economic meaning of a simulation outcome that changes when a technical parameter is slightly changed. Therefore in order to derive any results from evolutionary social simulation it is important that results are valid for a wider range of parameter settings, that is that they are robust. In this paper, the focus is on robustness with respect to the technical evolutionary algorithm parameters. The paper proceeds by giving a general overview of genetic algorithms in section 2 and a discussion of the interpretation of both genetic algorithms models for the learning behavior of economic agents in section 3. Section 4 describes the different experiments that were performed and conclusions are given in section 6.

2 Evolutionary algorithms

Evolutionary Algorithms. An evolutionary algorithm (EA) is a technique that uses the concept of "survival of the fittest" to evolve a population of strategies [25, 32]. Using an EA, strategies are represented as chromosomes and the chromosomes evolve from generation to generation yielding better and better strategies. A typical EA can be described as follows [32]. First, a population of randomly initialized chromosomes is generated. The population is subsequently changed and improved in a number of generations by means of selection, recombination ("crossover"), and mutation. Selection chooses the better chromosomes (with higher accumulated payoff) that will serve as parents for the next generation of strategies. This corresponds to the concept of "survival of the fittest" in nature. Offspring is then formed by pairwise recombination of the parents. Finally, the offspring chromosomes are slightly changed (with a small probability) and the new population replaces the old one.

Thus, an EA has several parameters, being among others the mutation rate (the probability for a gene or bit in a chromosome to change), the population size (the number of chromosomes in the population), the recombination rate (the percentage of chromosomes that will be subject to recombination) and the representation of the chromosome. The values of these parameters have to be carefully chosen in order for the EA to perform well. Although there are no conclusive results on which parameter settings allow effective EA performance, there are some general guidelines available (see for example [32, 9]).

Parameter issues. Particularly important in this paper is the avoidance of premature convergence by allowing the population diversity of the GA to be sufficient, that is a sufficient number of different elements (chromosomes) must be present in the population to avoid sampling errors due to small population size. This is a merely technical condition, under which the highly stylized EA operators (mutation, recombination, selection) and parameters operate properly. Premature convergence restricts the EA in it's learning capabilities. This occurs when, early in the search process, the EA focuses on the exploitation of a small selection of rather fit strategies at the expense of the exploration of other regions of the search space. In optimization problems premature convergence causes the EA to get stuck in various local optima (in different runs), while lacking the diversity to explore other regions of the search space and find the global optimum. To preserve sufficient diversity and to avoid premature convergence, the values of the EA-parameters have to be carefully chosen. Known factors that influence the diversity of the population (and thus the occurrence of premature convergence) are among others mutation rate, population size, selection pressure and the chosen chromosome representation. The length and representation of the chromosome affect convergence. The representation that is chosen should be able to represent all possible strategy values. Furthermore, we consider it good practice to choose chromosome length and representation in such a way that the average outcome of randomly initialized population of the chromosomes does not lie on or close to equilibrium values of the economic model.

3 Economic interpretation of genetic algorithms

Two general approaches to social or population learning can be distinguished and each approach represents a different economic interpretation of genetic algorithm learning. In the first approach each agent is represented by a single chromosome in the EA population (see for example [19, 42]). In this approach the number of chromosomes equals the number of economic agents. In the second approach the population of chromosomes and the economic model are separated (for example [7, 34]). The population of chromosomes is regarded as a pool of strategies from which the agents can choose. Both approaches consider the recombination operator as a model of information exchange between two agents or strategies. Whereas the mutation operator is regarded as a model for error or innovation. Although there thus exists an economic interpretation of the genetic operators the modeler should be very careful in interpreting the economic meaning of simulation results. It is very difficult to attach an economic meaning to social simulation results that were obtained using specific values for the GA parameters, in particular if outcomes change considerably with small changes of the parameters. In general, we propose that in order to make valid economic interpretations based on evolutionary simulations - results should be robust, that is valid for a larger range of parameter settings.

Several authors have commented on the economic interpretation of genetic algorithms [13, 19, 22, 27]. A commonly heard objection against this interpretation is that in population based evolutionary algorithms, strategies are shared by all agents. This means that agents have direct insight in which strategies were used by others. Furthermore, agents can copy (parts of) strategies from (well performing) others by means of selection and recombination. This is not true for many social systems; economic agents (firms, bargaining individuals), for example, often make great efforts to hide their strategies from their competitors. This suggests that the use of GAs as a model for agent learning is only valid when the strategies that are considered are simple actions that are observable in the market. The second modeling approach (chromosome equals a single strategy) is therefore preferred to the chromosome-equals-agent approach of population learning. Below we will illustrate this by showing the robustness results obtained for a large range of values.

4 Experimental setup

This section describes the experimental setup that was used to investigate the effect of evolutionary model parameter settings on economic outcomes. As described above two main approaches can be distinguished (see Table 1). Using approach I [19, 42], each agent is represented by a single strategy in the genetic population. The second approach allows an agent to pick a strategy from a large population of strategies. In approach I the value of the genetic algorithm parameter *population size* (that is the number of strategies in the genetic pool) is directly determined by the economic model. In approach II this is not the case. Using approach II [7, 34], we have to set the value for the population size in such a way that the economic outcomes are robust to changes of the parameter values (both economic and genetic).

Approach I	**Approach II**
Chromosome = Agent	Chromosome = Strategy
Population size determined by number of economic agents	*Population size determined by proper EA learning*

Table 1: Summarizing the two evolutionary learning approaches

The Cournot oligopoly The testbed for our simulations is a textbook [40, 18] Cournot oligopoly market with 4 players. We develop an evolutionary simulation of this model and use it to compare both approaches. The rest of this section will describe the Cournot oligopoly and the setup of the evolutionary simulation. We use a population learning evolutionary algorithm to model the search and learning behaviour of economic agents in a Cournot oligopoly game with four agents. The

Cournot oligopoly market describes a situation where a few firms compete in a single market. The firms have some market influence through a common price demand curve. In this situation, firms have to make a strategic decision, taking into account the decisions of the other firms. Here we focus on the Cournot oligopoly which provides a model for the market when the the firms produce a homogeneous good and compete on quantity. The 4 firm oligopoly model we use is characterized by the following equations:

$$Market\ price: \ P = 256 - Q, \ where\ Q \ = \sum_{i=1}^{4} q_i$$
$$Profit\ firm\ i: \ \pi_i = Pq_i - 56q_i$$

The Cournot-Nash equilibrium occurs when each firm's output is a best response to the combined output of the other firms - at output 40. When firms produce at the competitive outcome, each produces an output of (on average) 50 and the firms make no profit. Profits are highest when the 4 firms act as a single monopolist and produce the collusive output of on average 25 per firm. Collusion usually does not occur in the one-shot game.

A general outline of the Cournot oligopoly simulation is given below. Each period an agent draws a strategy from the population of strategies. If approach I is used the number of strategies is equal to the number of agents (4 in this paper). Using approach II we have to find a suitable population size (the number of strategies/chromosomes in the genetic population). In order to do this we incrementally (steps of 4) increase the population size and run the system repeatedly with a large number of values for the population sizes. Another parameter that influences simulation results is the representation of the strategies, in particular the length of the strategy. Longer strategies represent larger possible output values. Initializing the population with a representation that allows agents to produce quantities that exceed demand by far can be done if the modeler does not want the agents to have any a priori knowledge about the market (everything has to be learned). Again we look for settings that render robust results.

Outline of the simulation model:	
Step 1	**Select strategy:**
	Each agent randomly draws a strategy from the population of strategies
Step 2	**Determine action**
Step 3	**Play game**
Step 4	**Assign payoff**
	The fitness of a strategy equals the payoff gained by the agent using the strategy
Step 5	**If all strategies have been used:**
	Update strategies using EA

Table 2 gives an overview of the parameter settings used in the experiments. The smallest population size tested was 4 (i.e, equal to the number of agents - Approach I) and the largest population size tested 200. The chromosome length was varied between 6 (maximum output 64) and 9 (maximum output 512). In all cases the average output of a randomly initialized population does not lie on the Cournot-Nash or the competitive equilibrium. The structure of the initial population is determined by the *initial density* parameter. A population that is initialized with a density of 0.5 (the default value), fifty percent of the bits is set to 1 on average. The values for the mutation rate are within the range of commonly recommended values [24]. We consider the convergence of the model withe respect to the different outcomes of the game for a large range of values of the size of the population of strategies. Each period of the game agents select a strategy from the population of strategies. This population is subsequently updated by a genetic algorithm. The general outline of the simulation model is described below. A genetic algorithm is said to converge if almost all strategies in the population are identical. Note that there can be no convergence in the mathematical sense, for the mutations disrupt the uniform state over and over again.

	I	**II**
Economic Model Parameters		
Number of agents	4	4
EA Parameters		
Crossover rate	1.0	1.0
Mutation rate (per bit)	0.01	0.01
Population size	**4**	**4-200**
		(increments 4)
chromosome length	6-9	6-9
representation	binary	binary
initial density	0.1, 0.5 (default), 0.9	0.1, 0.5 (default), 0.9

Table 2: Parameter settings used in the Cournot oligopoly experiments. Approach I is the methodology where a chromosome equals an agent whereas in approach II a chromosome equals a strategy.

5 Robustness results - comparing the two approaches

This section describes the experiments we have performed comparing approach I and II to social learning. We are especially interested in the robustness properties of the two approaches. Apart from looking at the population size we study the behaviour of both approaches under different initial conditions and different representations. In all simulations we have used a simple GA with mutation rate 0.01 and crossover rate 1.0. First we study the effect of population size on the outcome of the Cournot oligopoly game. We start by initializing the population size at 4 (approach I) and then increasing this value (with steps of size 4) to determine a population size that yields robust results. Figure 1 shows the results of our experiments for different representations (chromosome lengths). The first figure (left) represents the relationship between average output and population size. Each point in the graph represents the average over 20 runs of 400 generations (generation 100-500 to compensate for initial noise effects) each that were executed using each value of the population size. The initial population density is 0.5 in these experiments, that is, initially half of the bits is set to one. The second figure (right) gives the standard deviation of average output over those 20 runs for each tested value of the population size. Results are given for different representations of the strategies, that is chromosome length 6,7, and 8.

5.1 Aggregate results

The outcomes for population size 4 correspond to approach I and we can deduce from the figure that population sizes of around 100 chromosomes would be suitable for approach II. Notice that for small values of the population size the difference between individual runs is very large (high standard deviation) whereas averages converge towards Cournot-Nash values for larger values. Convergence is slower for longer chromosome length where genetic drift causes averages to go up (as chromosome length increases a random mutation is more likely to lead to a large increase in production than with smaller chromosome lengths). Note that simulations using different representations all converge to the Cournot-Nash equilibrium with small standard deviation for sufficiently large population sizes. Furthermore, figure 1 suggests that convergence is not robust with respect to representation for small population sizes.

To gain more insight in the convergence behaviour of both approaches we take a look at the average population behaviour using approach I en II. Figure 2 shows the average population behaviour for different representations and different initial conditions over 500 generations. (Note

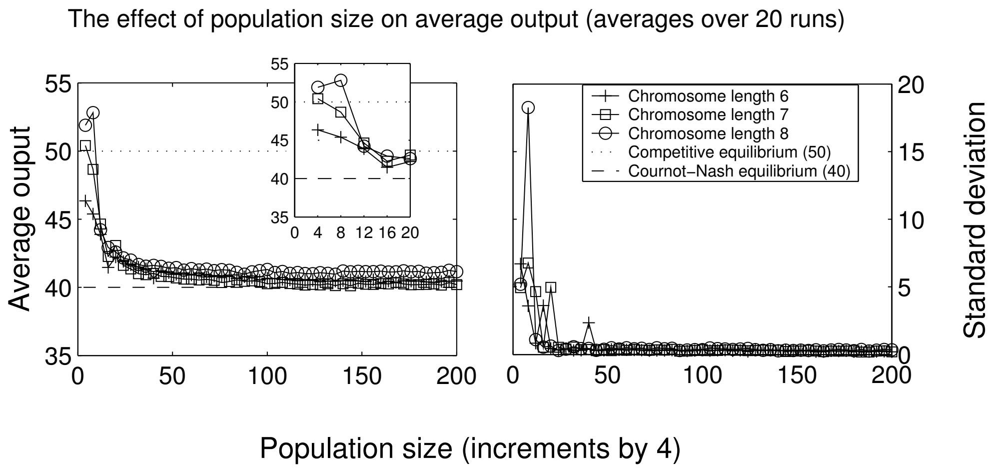

Figure 1: Average output versus population size, averages (left) and standard deviations (right) over 20 runs. Each plotted value represents the average output over 20 runs and over the last 400 generations (generation 100 to 500) to eliminate initial noise effects.

that the y-axis is larger for chromosome length 9 to accommodate all outcomes). All tested chromosome lengths allow the agents to produce at both the Cournot-Nash and competitive output levels and therefore do not constrain their production capacity in any way. The results show that approach I is not robust with respect to representation, that is different representations yield different economic outcomes. On the other hand population averages do converge towards Cournot-Nash outputs irrespective of representation length if approach II is used. If we look at convergence behaviour with respect to initial population density we see again that approach I does not yield robust results for all representations. Only for chromosome length 7 we can observe convergence towards the competitive equilibrium for all tested initial conditions. We will take a closer look at the individual runs in the next section to futher study the convergence behaviour. The outcomes in figure 2 suggest that the lack of robustness is caused by premature convergence caused by a lack of diversity in the population. Very small (0.1) or very large (0.9) initial densities lead to an almost uniform population and this effect is off course strongest for small population sizes. While some learning towards profitable outcomes occurs under these conditions, the population converges before a good outcome is reached. Indeed we see that outcomes for different representations show most agreement for initial density 0.5 when population diversity is maximized.

5.2 Individual runs

Figure 3 and Figure 4 show the twenty individual runs for each approach. Notice that there is no uniform convergence when approach I is used. More specifically, if we look at the results for chromosome length 7 in Figure 3 (bottom half) we see that the apparent convergence to the competitive equilibrium is only present at the aggregate level. Furthermore, we see that individual runs sometimes converge to very large output values (Figure 4) leading to losses for the agents. Here learning stops before profitable outcomes are reached. If approach II is used, however we see that all individual runs converge towards the Cournot-Nash level where is agents output is a best response to the aggregate output of the other agents. The agents obtain higher profits when their aggregate output is at Cournot-Nash level than at the competitive level. These results show that a significantly large population of chromosomes is needed to prevent premature convergence

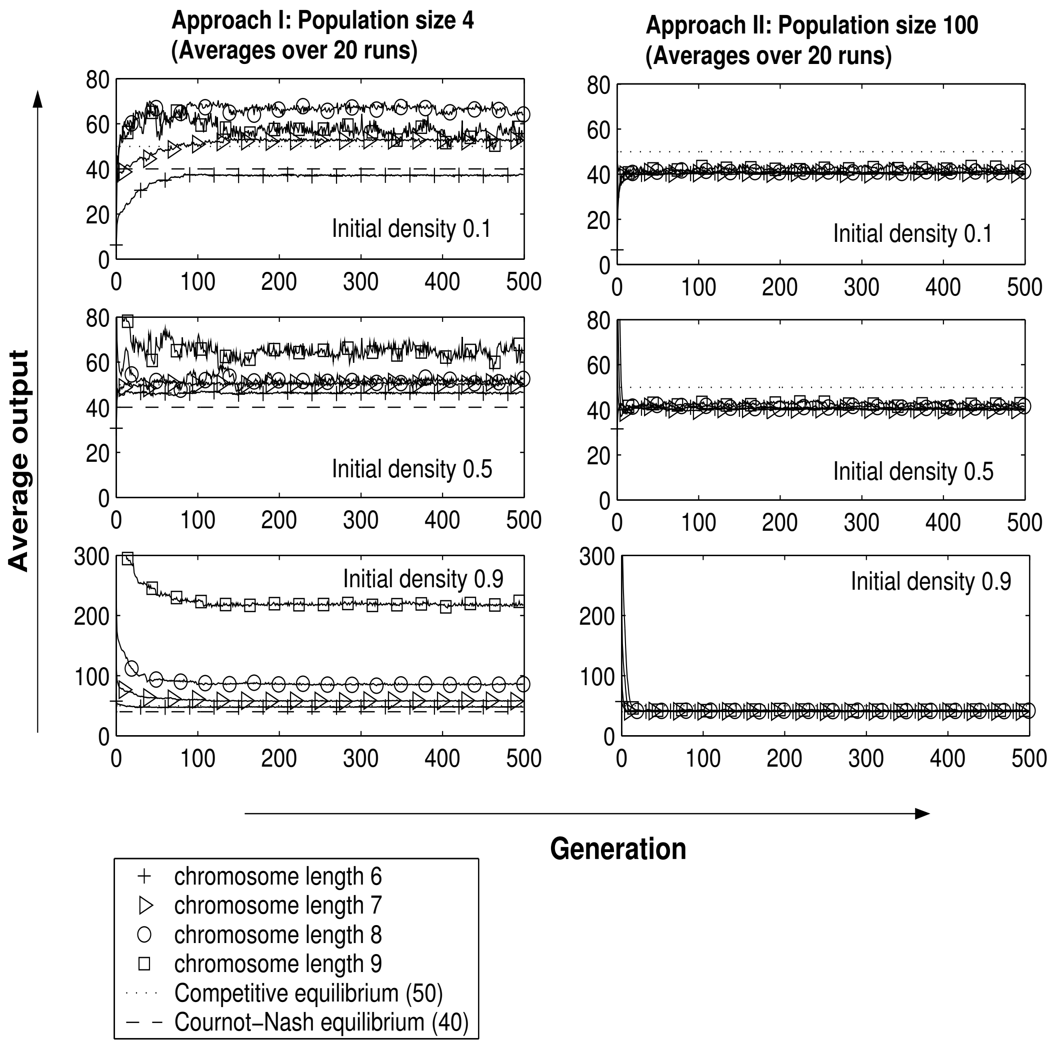

Figure 2: Average output for different population sizes - averages over 20 runs. Approach I: population size 4, Approach II population size 100. Note that the y-axis is larger for chromosome length 9 to accommodate all outcomes.

to a less profitable outcome. These experiments show that a sufficiently large population size is a necessary condition for convergence to the Cournot-Nash equilibrium in this model. Population learning genetic algorithms can thus be used to model procedurally rational agents in the Cournot oligopoly game. Evolutionary algorithm learning is a sufficiently powerful learning technique to arrive at the Cournot-Nash outcome in a population learning setting. When GA learning is inhibited due to small population sizes, the learning outcome depends strongly on the initial population and GA learning is reduced to imitation based learning. We have performed some experiments to illustrate this effect in the next section.

5.3 From imitation based learning to genetic algorithm learning

The evolutionary algorithm is built using selection, recombination and mutation. If we look at each of these components separately, we can gain further insight into the effects of population diversity. An algorithm using only selection (imitation) can be compared to replicator dynamics

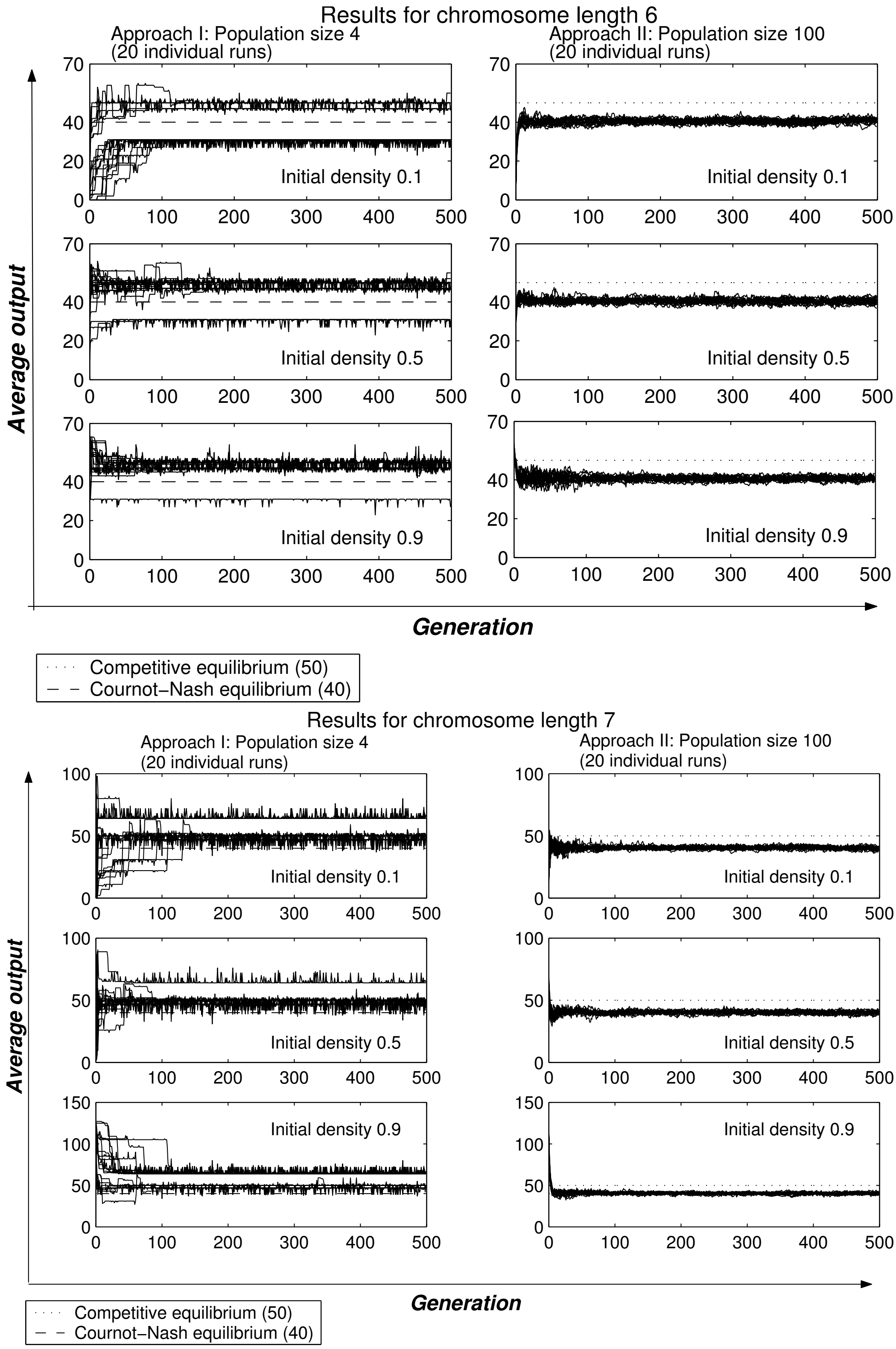

Figure 3: Individual runs for chromosome length 6 (top) and 7 (bottom). Note that not all axes are equal in order to accommodate all results.

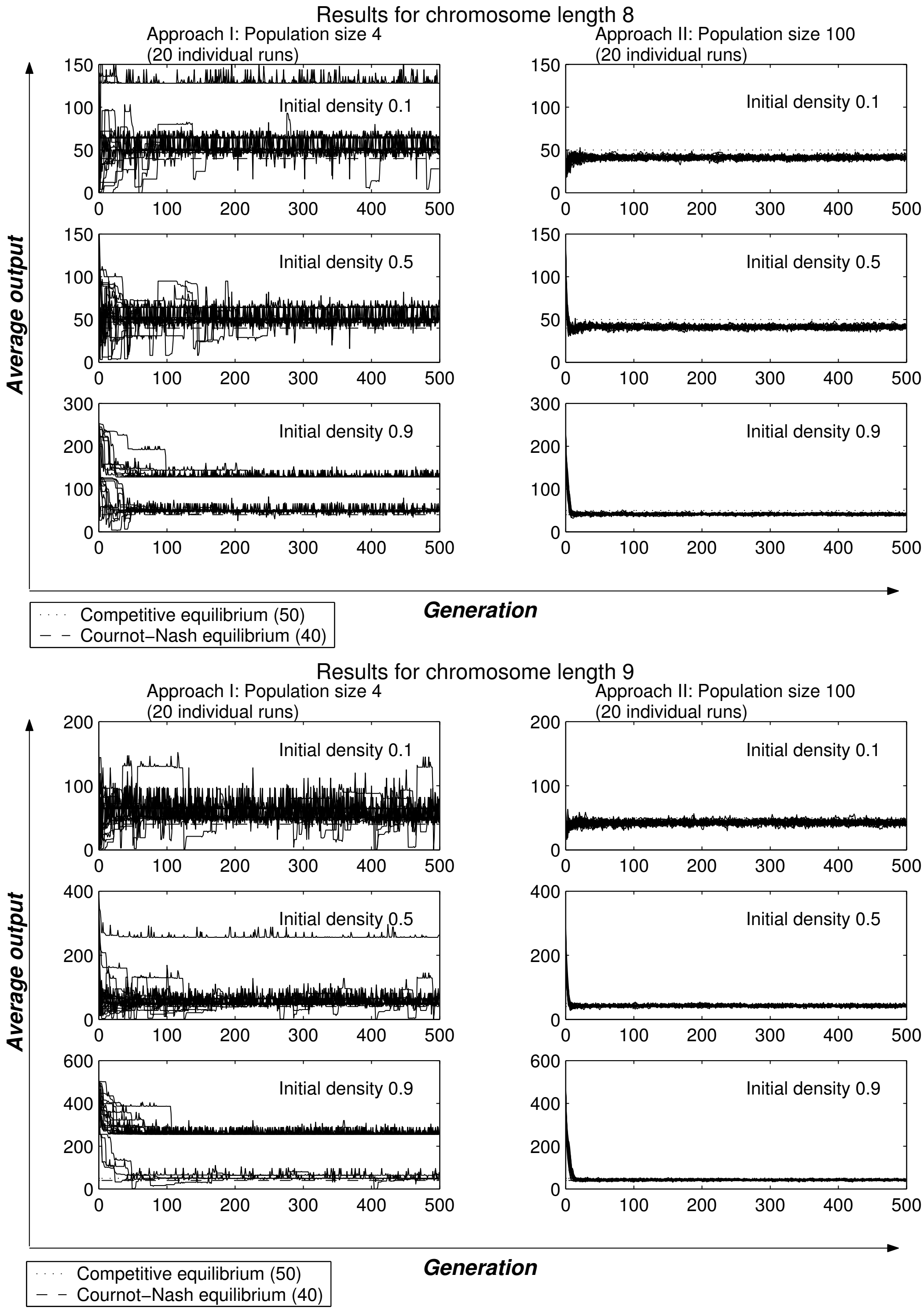

Figure 4: Individual runs for chromosome length 8 (top) and 9 (bottom). Note that not all axes are equal in order to accommodate all outcomes.

a simple learning model that is often used in economics [23, 43, 29]. The difference between replicator dynamics and genetic algorithm learning is that replicator dynamics is solely based on imitation while recombination and mutation are integral parts of genetic algorithm performance. Another difference between genetic algorithms and replicator dynamics is the way fitnesses are compared. In replicator dynamics, agents can imitate strategies used by a direct opponent while in a GA strategies used in all games are compared population wide. Theoretical results using replicator dynamics in oligopoly settings show that there is convergence towards the competitive equilibrium in the long run as long as the competitive outcome is present in the initial population. Replicator dynamics is based on imitation of more successful strategies. To illustrate the learning capabilities of a properly set up GA, we have extended a replicator dynamics version of the Cournot oligopoly with recombination and mutation. We have used the method of fitness comparison used in replicator dynamics (only imitate strategies of direct opponents). Furthermore, we have inserted the competitive outcomes in the initial population (as in [41]) . These experiments can give us some insight in how the recombination and mutation operator of the GA are influenced by small population sizes. Averages over 20 runs are given in Figure 5. Results are shown for only selection (imitation) and imitation plus recombination and mutation (mutation rate 0.1) for chromosome length 7. We see that convergence behaviour of the imitation based learner is less influenced by the population size than the GA. Using a sufficiently large population size GA convergence towards the Cournot-Nash outcome can be observed even when a relative fitness measure is used. This shows that approach II is indeed robust with respect to initial conditions, representation and also transformation of the fitness function. This is important since many economic applications are characterized by relative rather than absolute fitness measures.

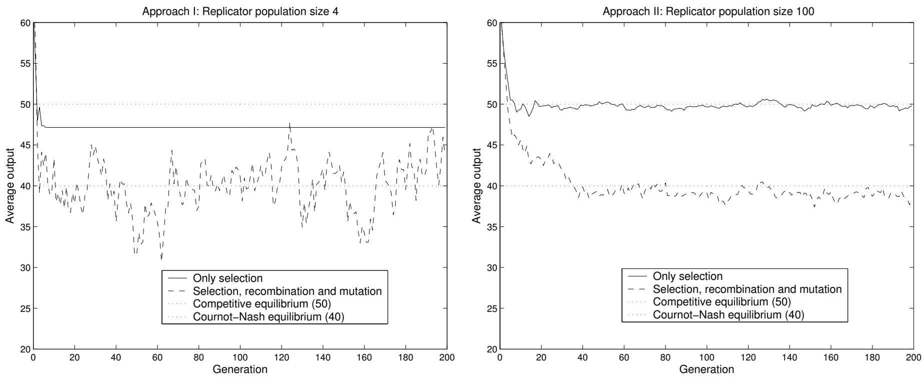

Figure 5: The influence of population size on imitation based and GA learning. Averages over 20 runs

6 Conclusions

We have compared two widely used approaches to evolutionary population learning. In the first approach population size is determined by the number of agents in the economic model while in the second approach these economic and technical parameters are treated separately. We show that the first approach can lead to premature convergence, inhibiting the learning capabilities of the agents, related to the abstract learning technique. The second approach leads to robust outcomes with respect to population size, initial density, representation of the strategies and fitness evaluation. These experiments show that economic model parameters and evolutionary algorithm parameters should be treated separately to allow for robust results. Furthermore, the modeler should make sure that simulation outcomes are robust with respect to changes in the experimental setup. This paper provides a first start in developing a robust methodology for the design and implementation of evolutionary economic simulations.

References

[1] N.M. Alemdar and S. Sirakaya. On-line computation of stackelberg equilibria with synchronous parallel genetic algorithms. *Journal of Economic Dynamics and Control*, 27(8):1503–1515, 2003.

[2] J. Arifovic. Genetic algorithm learning and the cobweb model. *Journal of Economic Dynamics and Control*, 18:3–28, 1994.

[3] J. Arifovic. The behavior of exchange rate in in the genetic algorithm and experimental economies. *Journal of Political Economy*, 104(3):510–541, 1996.

[4] J. Arifovic. Stability of equilibria under genetic algorithm adaptation. *Macroeconomic Dynamics*, 2(1):1–21, 1998.

[5] J. Arifovic. Evolutionary dynamics of currency substitution. *Journal of Economic Dynamics and Control*, 25(3–4):395–417, 2001.

[6] W.B. Arthur. On designing economic agents that behave like human agents. *Journal of Evolutionary Economics*, 3:1–22, 1993.

[7] R. Axelrod. The evolution of strategies in the iterated prisoner's dilemma. In *Genetic Algorithms and Simulated Annealing*, pages 32–41. Pitman, London, 1987.

[8] T. Bäck, D. Fogel, and Z. Michalewicz, editors. *Handbook of evolutionary computation*. Oxford University Press, 1997.

[9] T. Bäck, U. Hammel, and H.P. Schwefel. Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, 1(1):3–17, 1997.

[10] N. Basu. Aspen: A microsimulation model of the economy. *Computational Economics*, 12(3):223–241, 1998.

[11] J. Biethahn and V. Nissen, editors. *Evolutionary algorithms in management applications*. Springer Verlag, 1995.

[12] J. Bullard. Using genetic algorithms to model the evolution of heterogeneous beliefs. *Computational Economics*, 13(1):41–60, 1999.

[13] E. Chattoe. 'just how (un)realistic are evolutionary algorithms as representations of social processes?'. *Journal of Artificial Societies and Social Simulation*, 1(3), 1998. `http://www.soc.surrey.ac.uk/JASSS/1/3/2.html`.

[14] S.H. Chen, editor. *Genetic algorithms and genetic programming in computational finance*. Kluwer, 2003.

[15] S.H. Chen and C.H. Yeh. Towards a computable approach to the efficient market hypothesis: An application of genetic programming. *Journal of Economic Dynamics and Control*, 21(6):1043–1064, 1997.

[16] S.H. Chen and C.H. Yeh. Evolving traders and the business school with genetic programming: A new architecture of the agent-based artificial stock market. *Journal of Economic Dynamics and Control*, 25(3-4):363–393, 2001.

[17] C. Chiarella, M. Gallegati, R. Leombruni, and A. Palestrini. Asset price dynamics among heterogeneous interacting agents. *Computational Economics*, 22(2):213–223, 2003.

[18] A. Cournot. *Researches into the Mathematical Principles of the Theory of Wealth*. Macmillan, New York, 1897 (edited by N. Bacon, 1838).

[19] H. Dawid. *Adaptive learning by genetic algorithms.* Number 441 in Lecture Notes in Economics and Mathematical Systems. Springer Verlag, Berlin, 1996.

[20] H. Dawid. Learning of cycles and sunspot equilibria by genetic algporithms. *Journal of Evolutionary Economics*, 6:361–373, 1996.

[21] J. Duffy and P.D. McNelis. Approximating and simulating the stochastic growth model: Parameterized expectations, neural networks, and the genetic algorithm. *Journal of Economic Dynamics and Control*, 2001.

[22] B. Edmonds. Towards a descriptive model of agent strategy search. *Computational Economics*, 18:111–133, 2001.

[23] D. Fudenberg and D.K. Levine. Learning in games. *European economic review*, 42:631–639, 1998.

[24] J.J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Trans. on Systems, Man, and Cybernetics*, 16(1):122–128, 1986.

[25] J.H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Michigan, 1975.

[26] K.A. De Jong. *An analysis of the behavior of a class of genetic adaptive systems.* PhD thesis, University of Michigan, 1975.

[27] T.B. Klos. Decentralized interaction and co-adaptation in the repeated prisoner's dilemma. *Computational and Mathematical Organization Theory*, 5(2):147–165, 1999.

[28] C. Lawrenz and F. Westerhoff. Modeling exchange rate behavior with a genetic algorithm. *Computational Economics*, 21(3):209–229, 2003.

[29] G.J. Mailath. Introduction: Symposium on evolutionary game theory. *Journal of economic theory*, 57:259–277, 1992.

[30] R. Marimon, E. McGrattan, and T. Sargent. Money as a medium of exchange in an economy with artificially intelligent agents. *Journal of Economic Dynamics and Control*, 14:329–373, 1990.

[31] D.F. Midgley, R.E. Marks, and L.G. Cooper. Breeding competitive strategies. *Management Science*, 43(3):257–275, 1997.

[32] M. Mitchell. *An introduction to genetic algorithms.* MIT Press, Cambridge (MA), 1996.

[33] T. Riechmann. Genetic algorithm learning and evolutionary games. *Journal of Economic Dynamics and Control*, 25(6-7):1019–1037, 2001.

[34] T. Riechmann. *Learning in Economics.* Physica-Verlag, Heidelberg (Germany), 2001.

[35] E. Ringhut and S.Kooths. Modeling expectations with genefer - an artificial intelligence approach. *Computational Economics*, 21(1):173–194, 2003.

[36] S. Schulenburg and P. Ross. An adaptive agent based economic mode. In P.L. Lanzi, W. Stolzmann, and S.W. Wilson, editors, *Learning Classifier Systems: From Foundations to Applications*, volume 1813 of *Lecture Notes in Artificial Intelligence*, pages 265–284. Springer, 2000.

[37] L. Tesfatsion. Introduction to the special issue on agent-based computational economics. *Journal of Economic Dynamics and Control*, 25(3-4):281–293, 2001.

[38] E.P.K. Tsang and J. Li. Combining ordinal financial predictions with genetic programming. In K.S. leung, L.W. Chan, and H. Meng, editors, *Proceedings of the second international conference on intelligent data engineering and automated learning (IDEAL2000)*, volume 1983 of *Lecture Notes in Computer Science*, pages 532–537. Springer, 2000.

[39] P. Tuson and P. Ross. Adapting operator settings in genetic algorithms. *Evolutionary computation*, 6(2):161–184, 1998.

[40] H.R. Varian. *Microeconomic analysis*. W.W. Norton, New York, 1992.

[41] F. Vega-Redondo. The evolution of walrasian behavior. *Econometrica*, 65:375–384, 1997.

[42] N.J. Vriend. An illustration of the essential difference between individual learning and social learning and its consequences for computational analyses. *Journal of Economic Dynamics and Control*, 24:1–19, 2000.

[43] J. Weibull. *Evolutionary Game Theory*. MIT Press, Cambridge (MA), 2002.

[44] M. Yildizoglu. Competing r&d strategies in an evolutionary industry model. *Computational Economics*, 19(1):51–65, 2002.