DOES BOUNDED RATIONALITY LEAD TO INDIVIDUAL HETEROGENEITY?

THE IMPACT OF THE EXPERIMENTATION PROCESS

AND OF MEMORY CONSTRAINTS

Marco Casari[1]

Universitat Autònoma de Barcelona

UFAE and IAE Working Paper number 583.03

Abstract:

In this paper we explore the effect of bounded rationality on the convergence of individual behavior toward equilibrium. In the context of a Cournot game with a unique and symmetric Nash equilibrium, firms are modelled as adaptive economic agents through a genetic algorithm. Computational experiments show that (1) there is remarkable heterogeneity across identical but boundedly rational agents; (2) such individual heterogeneity is not simply a consequence of the random elements contained in the genetic algorithm; (3) the more rational agents are in terms of memory abilities and pre-play evaluation of strategies, the less heterogeneous they are in their actions. At the limit case of full rationality, the outcome converges to the standard result of uniform individual behavior.

JEL Classification numbers: C72, D83, C63

Keywords: bounded rationality; genetic algorithms; individual heterogeneity

---

[1] Correspondence address: Marco Casari, Departament d'Economia i d'Historia Económica, CODE, Edifici B, Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain, email: mcasari@pareto.uab.es, tel: ++34.93.581 4068, fax: ++34.93.581 2461.

The paper has benefited from discussions with Jasmina Arifovic, Simon Wilkie, Charles Plott, Nelson Mark, Sean Gailmard, Guillaume Frechette, Ben Klemens and the comments of participants at seminars at Ohio State University, the 5th Workshop in Experimental Economics in Siena, the ESA meeting in San Diego, CA, USA and the FUE Summer School in Bounded Rationality in San Sebastian, Spain. Sharyn Slavin Miller, Maria Satterwhite, and Eloisa Imel from Caltech provided technical support. Financial support from the Division of the Humanities and Social Sciences at Caltech and the EU Marie Curie Fellowship is gratefully acknowledged.

DOES BOUNDED RATIONALITY LEAD TO INDIVIDUAL HETEROGENEITY? THE IMPACT OF THE EXPERIMENTATION PROCESS AND OF MEMORY CONSTRAINTS

1 INTRODUCTION

Common assumptions in economics are that decision makers have unlimited time, information processing capabilities, and that they can be conceptualized as Bayesian maximizers. In this paper we restrict some of these assumptions within a genetic algorithm framework and focus the attention on individual convergence to equilibrium. We show that where identical and fully rational agents behave in a uniform manner, identical - but boundedly rational - agents can each behave differently.

In the context of a Cournot game, we tackle the issues of the impact of limited information processing capabilities on individual firms' heterogeneity. The firms are modeled as adaptive economic agents with limited knowledge of the task and limited memory. They start with different strategies, which have been assigned to them randomly, they experiment with new strategies, and they learn from experience. In order to implement an evolutionary approach focused on the individual firm, we employ a genetic algorithm framework, where each agent is endowed with a separate set of strategies. This class of genetic algorithms is called multi-population or individual learning because the agents learn from their own experience, in contrast with the other class, known as single-population or social learning, where agents learn from other agents' experiences (Dawid, 1999; Holland and Miller, 1991; Vriend, 2000; Chen and Yeh, 2001). The social learning architecture has been successfully employed in the agent-based computational literature in economics to study aggregate behavior (Bullard and Duffy, 1998; Miller, 1996; Arifovic, 1996; Nowak and Sigmund, 1998) but we argue that the individual learning architecture is better suited for this study because it focuses on the individual behavior of the agents (Andrews and Prager, 1994; Arifovic, 1994; Chen and Yeh, 1998).

Simulation results show that boundedly rational agents exhibit a remarkable heterogeneity in behavior. Contrary to what one might expect, this result is not simply a consequence of the random elements contained in the genetic algorithm. For instance, consider agents that randomly draw the strategy they play from a uniform distribution every period. We show that the resulting individual heterogeneity of such agents that play completely at random is lower than from the interaction of genetic algorithm agents. Moreover, with a rise in the memory capabilities and the ability to explore unavailable strategies, individual differences decline and the results suggest that, in the limit case of full rationality, we obtain the standard result of uniform individual behavior.

The paper is organized as follows. In Section 2 we outline the Cournot model and the parameter values used in the simulations. Then, the procedures followed by agents in making decisions are explained in Section 3, where we illustrate the design of the genetic algorithm and state some of its properties. The main result about individual heterogeneity is in Section 4, along with the discussion about whether it is simply due to excessive noise. Our findings suggest that it is not. Moreover, the higher the rationality level of the decision-makers, the lower the level of individual heterogeneity. Changes in rationality levels are generated by adjusting pre-play evaluation of new strategies through a weaker and a stronger filter (trembling hand and election, in Section 5) and by varying working memory size (Section 6). Conclusions are in Section 7.

2 THE COURNOT EXAMPLE

The playground for the boundedly rational agents is a standard Cournot oligopoly game, $\Gamma(N, (S_i)_{i\in N}, (\pi_i)_{i\in N})$. In the game, there are N identical firms who all compete in the same market and produce the same homogeneous commodity. The decision variable for firm *i* is the quantity x_i to be produced, which lies in $[0, \vartheta]$. All firms simultaneously choose a production level, and then a market price *p* is determined through the clearing of market demand and supply. Let us assume that the inverse demand function is $p(X)=d-bX$, where $X=\sum_{i=1}^{N} x_i$ and $d, b>0$; and the cost function is $c(x_i)=\alpha x_i$, which is linear and identical for all firms. Hence the profit function is

$\pi_i = x_i\left(d - bX\right) - \alpha x_i$. Firms are profit maximizing and face the same incentive structure for T interactions without carry-over from one period to the next.

The Nash equilibrium of the game is $X^* = \frac{N}{(N+1)} \cdot \frac{d-\alpha}{b}$ and p*=$\frac{1}{(N+1)}\left(d - N\alpha\right)$. The game has a continuous strategy space and a unique, symmetric, and evolutionary stable Nash equilibrium. In other words, this game provides ideal conditions to facilitate convergence toward the Nash equilibrium outcome both at the aggregate and individual levels. The parameter values adopted are N= 8, ϑ=50, α = 5/2, d = 23/2, b=1/16, which yields Nash equilibrium values of X*=128, x_i*=16 ∀i, and p*=7.5.[2] At the Nash equilibrium, industry profits are less than monopoly profits; in particular, earnings are 39.5% of monopoly profits. As will be explained in the next Section, the precise numerical values of the parameters are not crucial for these analyses. More precisely, any positive monotonic transformation of the payoff function will not alter the results in this paper (Proposition 2). The Cobweb model is a very common playground in the computational economics literature and different specifications for price expectations have been put forward (Hommes, 1998; Jensen and Urban, 1984; Arifovic, 1994). In his seminal work, Ezekiel (1938) employs naïve expectations where current price expectations are simply the last period price, $p_t^e = p_{t-1}$. In the Cournot setting adopted here, price expectations for the current period are equal to the price in the last period adjusted by variations due to changes in the quantity that the firm itself is going to produce, $p_t^e = p_{t-1} - b\left(x_{i,t} - x_{i,t-1}\right)$.

3 DESCRIPTION OF THE GENETIC ALGORITHM AGENTS

Genetic algorithm agents will play the Cournot game described in the previous Section. A full description of the working of a genetic algorithm (GA) is given in the textbooks of Holland (1975), Goldberg(1989), and Mitchell (1996). For issues specific to Economics see the excellent study of Dawid (1996). This section introduces the GA decision makers along with the parameter values

[2] Notice that the Nash Equilibrium outcome is not positioned in the center of the action space (i.e. at 200) so that it would not be reached through pure chance by zero intelligence agents.

used in the simulations. In order to analyze individual behavior, the simulations employ an individual learning (multi-population) genetic algorithm. Three interesting aspects of this framework are discussed in more detail: the memory set, the choice rule, and the ordinal nature of the GA.

The genetic algorithm decision maker can be described as follow. The strategy which agents have to choose is identified by a single real number. It is encoded as a binary string, a so-called chromosome, and has associated with it a score (measure of fitness) that derives from the actual or potential payoff from this strategy. In a social learning (single-population) basic GA, each agent has just one strategy (chromosome) available, which may change from one period to the next. The changes are governed by three probabilistic operators: a reinforcement rule (selection), which tends to eliminate strategies with lower score and replicate more copies of the better performing ones; crossover, which combines new strategies from the existing ones; and mutation, which may randomly modify strategies.[3] In a basic GA, the strategies (chromosomes) created by crossover and mutation are directly included in the next period's set of strategies (population).

The three operators are stylized devices that are meant to capture the following elements involved in human learning when agents interact. The reinforcement rule (selection) represents evolutionary pressure that induces agents to discard bad strategies and imitate good strategies; crossover represent the creation of new strategies and the exchange of information; mutation can bring new strategies into a range that has not been considered by the agents.

Simulations are run with an individual learning GA (Figure 1), which is discussed in the remainder of this section. When agents do not consider just one strategy at each period in time, but have a finite collection of strategies from which one is chosen in every period (memory set), the process is called a multi-population GA. A strategy is a real number $a_{ikt} \in [0,50]$ that represent the production level of firm i in period t. Each agent is endowed with an individual memory set $A_{it}=\{a_{i1t}, \ldots, a_{iKt}\}$ composed of a number of strategies K that is constant over time and exogenously

[3] The crossover operator first, randomly selects two strategies out of a population; second, selects at random an integer number w from [1, L-1]. Two new strategies are formed by swapping the portion of the binary string to the right of the position w.

given. If a strategy a_{ikt} is in the *memory set*, i.e. it is available, agent i can choose it for play at time t. When there exist more than K strategies in the game, there are always strategies that are not currently available in the memory set. Notice that an available strategy has no impact on the outcome unless it is chosen for play.

The size of the memory set, *K*, is a measure of the level of sophistication of an agent since it determines how many strategies an agent can simultaneously evaluate and remember. The Psychology literature has pointed out that the working memory has severe limitations in the quantity of information that it can store and process. According to these findings, the memory limitation is not just imperfect recall from one round to the next, but rather an inability to maintain an unlimited amount of information in memory during cognitive processing (Miller, 1956; Daily et al., 2001). By setting K=6 we assume that decision-makers have a hardwired limitation in processing information at 6 strategies at a time. The classic article by Miller(1956) stresses the "magic number seven" as the typical number of units in people's working memory.[4]

As each agent is endowed with a memory set, in the multi-population GA there is an additional issue of how to choose a strategy to play out of the K available. The *choice rule*, C: **A(K)** $\rightarrow$ **A**, is a stochastic operator that works as a one-time pairwise tournament, where (1) two strategies, a_{ikt} and a_{iqt}, are randomly drawn with replacement from the memory set A_{it} and (2) the strategy with the highest score in the pair is chosen to be played: $a^*_{it}=\text{argmax}\{s(a_{ikt}), s(a_{iqt})\}$. A pairwise tournament is different from deterministic maximization, because the best strategy in the memory set is picked with a probability less than one.[5] The choice rule, however, is characterized by a probabilistic response that favors high-score over low-score available strategies. In particular, the probability of choosing a strategy is strictly increasing in its ranking within the memory set (Proposition 1). The

---

[4] The memory set size K needs to be even, so it could have been set to 8. There is debate in the psycological literature about what constitutes an unit when counting to 7. In this specific application it seems reasonable to identify a single strategy as a unit.

[5] In general, an M-tournament choice rule is weaker than a deterministic maximization ("choose the available strategy with the highest score") in two ways. First, the number of available strategies involved in the tournament is generally lower that the size of the memory set, M<K, and so only a subset of available strategies is actually compared (with a pairwise tournament, M=2). Second, even when M=K, the choice rule is different from deterministic maximization because the M available strategies are drawn with replacement. More precisely, there is a $[(K-1)/K]^K$ chance of choosing a strategy different from the best one in the set.

stochastic element in the choice captures the imperfect ability to find an optimum, where the probability of a mistake is related to its cost.

Proposition 1: The probability that an available strategy x is chosen for play, x*=x, by a pairwise tournament choice rule out of a set A of K available strategies is $P\{x^* = x\} = \frac{2r_x - 1}{K^2}$, where r_x is the ranking of the available strategy x within the set A (the worst available strategy ranks 1, r_x=1, and there is an assumption that there are no ties). ♦ Proofs to propositions are in the Appendix.

The most common choice rules in the literature are pairwise tournament and biased roulette wheel. We have adopted a pairwise tournament because it is ordinal, in the sense that the probabilities are based only on "greater than" comparisons among strategies. While in a biased roulette wheel the score needs to be positive and its absolute magnitude is important to compute the probability of being replicated, none of these matter for an ordinal operator like pairwise tournament. An ordinal operator does not rely on a "biological" interpretation of the score as a perfect measure of the relative advantage of one strategy over another (Proposition 2).

Proposition 2: The results of the GA agent interactions are unaffected by any strictly increasing transformation v:R→R of the score function. ♦

The score is the index of performance for a strategy a_{ikt} and is a function of the monetary payoff π, $s(a_{ikt}) = v[\pi(a_{ikt}, a_{-it})]$. The score of a strategy can be interpreted as the utility of the outcome associated with that strategy. Given the ordinality of pairwise tournaments adopted for reinforcement and choice rule, this GA is based only on the ordinal information of the score, like the utility function of the consumer. As a consequence, the simulation results are robust to any strictly increasing payoff transformation v.

A score is assigned to every strategy in the memory set, whether the strategy was chosen to be played or not. The distinction between the two cases is conceptually rather important. Assigning a score to a strategy that was actually employed (*actual* score) is an instance of reinforcement learning. Assigning a score to all the other available strategies, which were not actually used (*potential* score), always relies on a model, however subjective and imperfect, of the behavior of the other agents (Kreps, 1998). As already explained in Section 2, the model assumes that the price expectation in the current period is equal to the price in the last period adjusted by variations due to changes in the quantity that the firm itself is going to produce: $p_t^e = p_{t-1} - b\left(x_{i,t} - x_{i,t-1}\right)$.[6]

4 INDIVIDUAL HETEROGENEITY

Before proceeding to outline the simulation results, an example is presented to introduce the precise definition of individual heterogeneity adopted throughout the paper. After stating the main conclusion (Result 1), some benchmark cases are provided to show that noise, which is built into a GA, is not responsible for the claim of individual heterogeneity across agents (Result 2).

The same level of aggregate variability can hide widely different patterns of individual variability. The following example illustrates which is the individual dimension that matters for our analyses. Consider scenarios A and B in Table 2 with two players and four periods.

The two scenarios are identical when considering both aggregate production $X_t=\Sigma_i\, x_{it}$ and overall indexes of variability of individual actions, such as the mean of the difference, period by period, between the maximum and minimum individual productions, D1=$\frac{1}{T}\sum_{t=1}^{T} \max_i \left\{x_{it}\right\} - \min_i \left\{x_{it}\right\}$, or the standard deviation of individual actions x_{it} (SD1). The differences in the patterns of individual variability between scenario A and B can be captured by splitting the overall individual variability into variability across agents (D2 and SD2) and over time (SD3). In order to calculate agent-specific variability, first we compute the average individual production over time $\overline{x_i} = \frac{1}{T}\sum_{t=1}^{T} x_{it}$ and, using those data, compute the difference D2=$\max_i \left\{\overline{x_i}\right\} - \min_i \left\{\overline{x_i}\right\}$ and the standard deviation for $\overline{x_i}$ (SD2)

---

[6] From the previous discussion it is apparent that the knowledge and computational abilities assumed for a GA agent are very limited. An agent should be able to (1) count from 1 to K, (2) flip coins, (3) make ordinal comparisons between two real numbers, (4) evaluate the score of an outcome, and (5) remember K strategies. The toughest requirement comes for the assignment of the potential score, where an agent needs also (6) to understand how the outcome of the last period changes when he adopts a different strategy while everybody else does not. Agent i must know his payoff function, π_i, and others' aggregate actions from the previous period, a_{-it}. He does not need to know the payoff function of other agents, π_j with $j\neq i$, their individual past strategies, a^*_{jt}, whether they are fully or boundedly rational players, nor how many agents N there are.

(Table 2). Scenario A rates highly in terms of variability across agents, and that is referred to here as high individual heterogeneity, while scenario B rates highly in terms of variability over time but exhibits no individual heterogeneity.

When the same statistics developed for the example in Table 2 are applied to the simulation results (Table 3), a remarkable level of heterogeneity emerges from the interaction of ex-ante identical genetic algorithm agents (Result 1).

Simulation result 1 (Individual heterogeneity)

In a game with a unique Nash equilibrium, boundedly rational agents (multi-population genetic algorithms) with identical preferences and identical rationality levels generate individual behavior that is heterogeneous across agents. ♦

The interaction of GA agents generates a difference between minimum and maximum of D2=11.08, which constitutes 69% of the individual symmetric Nash equilibrium outcome of x_i*=16 and 22% of the range [0,50] of the individual strategy space. The standard deviations of individual production averages is SD2=3.68 (column (2) in Table 3). All of the results in Table 3 are averages over 100 runs with different random seeds. A single run consists of 400 iterations among the agents and the results reported are relative to the last 100 iterations (from 301 to 400).

Although bounded, the agents are endowed with identical levels of rationality. Yet they generate individually distinct behavior. Had they been designed with differentiated goals or variable skills, the heterogeneity of behavior would have not been surprising. When in experimental data identical incentives are given and heterogeneous behavior is observed (Ledyard, 1995, p.170-173; Casari and Plott, forthcoming), the explanation generally put forward is an individual-specific utility function.

The only built-in diversity in the genetic algorithm agents is the random initialization of the strategies. In other words, agents do not have common priors. Besides random initialization, there are four other stochastic operators (reinforcement rule, choice rule, crossover, mutation) that might introduce variability in the data. In order to have a benchmark to evalutate the influence of randomness and the magnitude of individual heterogeneity, the GA outcome can be compared with the result of interactions among zero intelligence agents ((2) vs. (8) in Table 3).

Zero intelligence agents are designed in the spirit of Gode and Sunder (1993) and are essentially pure noise,[7] as the individual strategy for each firm is drawn from a uniform distribution on the strategy space [0,50] and then aggregated to compute market production and price. Individual heterogeneity of zero intelligence agents is remarkably lower than in the case of genetic algorithm agents. While scoring much higher in terms of overall variability ($D1_{ZI}$=39.09 vs. $D1_{GA}$=15.06), zero-intelligence agents are characterized by half as much individual heterogeneity as genetic algorithm agents ($D2_{ZI}$=4.17 vs. $D2_{GA}$=11.08, $SD2_{ZI}$=1.41 vs. $SD2_{GA}$=3.68). In other words, Result 1 is not a consequence of the noise built into the GA.

In interpreting the results, it might be helpful to illustrate individual heterogeneity in outcomes with the following example. Suppose that every day you parachute a person from an airplane into the same unfamiliar region with the goal of reaching a specific point by foot. You give to the person a detailed area map. If all agents can read a map and have good orienteering skills, they will converge to the agreed upon spot at the end of the day. Now, suppose that you don't give the map but just show it before take off. Even under the restrictive assumptions that everybody can remember the same quantity of information from the map, one might wonder whether everybody will be at the same spot at the end of the day. It may depend on the specific features that each of them have memorized and on the way in which they assess the success of a new path.

Among the stochastic operators of a GA, consider the innovation process, and in particular the mutation rate, which is the prime source of noise. The composition of the memory set changes, among other reasons, because of active, random experimentation. In this study the strategy space is divided into a grid and strategies expressed in real numbers in the decimal system are translated into equivalent binary strings of 0s and 1s. This paper follows the uniform binary mutation process at the rate *pm*, which is common in the GA literature. Under this innovation process, there is a

---

[7] In Gode and Sunder(1993) they are subject to a budget constraint as well.

probability $pm \in (0,1)$ that each digit '0' flips to '1' or vice versa, with *pm* held constant for all the digits irrespective of their high or low cardinality of the string. Hence, the transitions from 111 to 101 and from 111 to 011 happen with the same probability.

Given a mutation rate *pm*, for ease of interpretation we can translate it into an *innovation level p* – which measures the expected percentage of strategies that will change because of the innovation process (mutation) – using the formula $p=1-(1-pm)^L$, where L is the number of digits of the binary string. A value of *pm*=0.02 such as the one adopted in the baseline GA corresponds to an expected number of new strategies due to innovation of 14.92% of the total in the memory set.[8]

A comparative static analysis is performed to evaluate the impact of a different innovation level. Two cases of special interest are when the innovation level moves toward zero and when it moves toward one.

Before presenting the impact of the innovation rate on individual heterogeneity, a clarification is necessary regarding the reinforcement operator (selection). The level of variability in the outcome is the result of two opposite tendencies at work within the genetic algorithm. One is the generation of new strategies because of the innovation operator and the other is the elimination of bad strategies due to the reinforcement operator. Hence, an excessive variability might be due more to a high innovation rate than to a weak reinforcement operator. As it will be explained, in this GA design the opposite is true because a pairwise tournament implements a stronger selection than a biased roulette wheel.

The reinforcement rule (selection) is a pairwise tournament repeated K times, R: $\mathbf{A(K)} \rightarrow \mathbf{A(K)}$, which is applied separately to each agent's memory set: (1) at time t two strategies, a_{ikt} and a_{iqt}, are randomly drawn with replacement from the memory set A_{it} and (2) the strategy with the highest score in the pair is placed in the new set: $a_{i\bullet t+1}=\text{argmax}\{s(a_{ikt}), s(a_{iqt})\}$; (3) the previous two

[8] The innovation rates used in four other studies are the following: Arifovic (1996) uses L=30 and pm=0.0033 or pm=0.033, which translates into p=0.0944 or p=0.6346; Andreoni and Miller (1995) L=10, pm=0.08 with exponential decay and half-life of 250 generations: p=0.5656 for t=1 and 0.0489 for t=1000; Bullard and Duffy (1998), L=21, pm=0.048: p=0.6441; Nowak and Sigmund (1998), p=0.001.

operations are performed K times in order to generate a complete memory set for agent i at time t+1, $A_{i,t+1}$.

Agents are adaptive learners in the sense that successful strategies are reinforced. Strategies that perform well – or that would have performed well if employed – over time gradually replace poor-performing ones. With experience, the composition of the memory set becomes the distilled wisdom of past decisions and past outcomes.

A key characteristic of a reinforcement rule is how quickly a successful strategy displaces the others in the memory set. One measure of this is the expected takeover time (TOT), which indicates how many iterations of the reinforcement rule it takes (in expectation) for a new strategy that has the highest score in the set to replace all other strategies in the set (Bäck, 1996). At the end of the process, when no other new strategy is introduced, all strategies are copies of the successful new one.[9] The takeover time of the pairwise tournament rule is TOT=(ln K + ln(ln K))/ ln 2, which equals 3.43 iterations for a memory set of size 6 (Bäck, 1996).

According to Bäck(1996), the expected takeover time with biased roulette wheel reinforcement when the score function is exponential, f(x)=exp(cx), is approximately (1/c) K ln K, which is of order o(K ln K). Given that pairwise tournament TOT is of order o(ln K), for large memory sizes tournament rules discard bad strategies faster than biased roulette wheel rules.[10-11]

After this clarification, let us turn to the effect of the innovation rate on the individual heterogeneity stated in Result 1. The results of varying p from 0.025 to 0.992 (pm=0.005-0.45) on the variability indexes D1, D2, SD2, SD3, X, SD(X) are shown in Figure 2 The data shows four

---

[9] A shorter takeover time is not necessarily better, though, because keeping the knowledge of old available strategies can be useful when the "environment" changes. Suppose for instance that there is an exogenous 6-period-long cycle in the environment and that there are only two possible strategies: x, best for the first 3 periods, and y, best for the last 3 periods. If TOT=2 the agent will lose memory of one strategy and needs to learn it all over again by experimentation at every cycle. An agent with TOT=4 will perform better.

[10] The same inequality holds for small numbers under mild conditions. For instance, for every K>2 when c<2.

[11] Two additional comments on the comparison between reinforcement rules in a genetic algorithm and in a replicator dynamic: first, in evolutionary game theory the replicator dynamic works in a context similar to the single-population genetic algorithm environment (social learning), while here the architecture is of multi-population GA (individual learning). In our design, there is no imitation of strategy from one agent to another (i.e., across different sets). Second, replicator dynamics generally work in continuous time while a genetic algorithm works in discrete time. It is shown in Weibull (1995) that in discrete time strictly dominated strategies need not to get wiped out as they are in continuous time.

major results. First, GA agents are no less individually heterogeneous than zero-intelligence agent for all innovation rates. Second, as the innovation rate approaches zero, individual heterogeneity does not disappear, on the contrary, it is at its highest peak ((3) in Table 3). Third, as the innovation rate grows, individual heterogeneity declines toward the level of zero-intelligence agents and, at the same time, variability over time steadily grows. Fourth, for higher levels of noise, especially beyond p=0.30, the aggregate outcomes moves away from the aggregate Nash equilibrium outcome. These considerations leads us to state Result 2.

Simulation result 2 (Heterogeneity and randomness) The high level of innovation of the agents is not responsible for the individual heterogeneity result. ♦

5 EFFECTS OF THE ELECTION OPERATOR

Identical bounded rationality agents produce individually heterogenous outcomes (Result 1). Does this result depend on the modality of evaluation of new strategies? Or on the memory constraints? This section and the next one look at what aspects of bounded rationality are responsible for the main result by exploring two dimensions of the rationality of GA agents, the process of pre-play evaluation of new strategies and working memory constraints. We begin with the former dimension.

A GA agent is characterized not only by its level of innovation but also by the filters that exist between the creation of a new strategy and the decision to choose it for play. In the baseline agent design, all new strategies are assigned a *potential* score before one strategy is chosen from the memory set (Fresh score).[12] Two other new strategy evaluation designs are now discussed: a weaker filter (trembling hand) and a stronger filter (election operator).

In the trembling hand design a new strategy keeps the score of its parent strategy. The “parent” strategy is the original strategy before the mutation happened or, in case of crossover, the strategy that has determined the highest bits in the binary string of the new one. The behavioral interpretation of trembling hand is of an agent that does not realize that a new strategy is different

[12] The score is only potential because the new strategy was not used for play (see discussion in Section 2).

from the parent strategy until the following periods. As a consequence he could play it, with the intention of playing the parent strategy and assigning to it a new score afterwards.

The election operator screens each new strategy before it is permitted to become an available strategy for play. This operator has become more and more common in social science applications (Arifovic, 1994; Bullard and Duffy, 1998). The new strategy replaces its parent strategy in the memory set only if its potential score improves its parent's (potential or real) score. If the score of the new strategy is lower than its parent strategy's score, the parent strategy remains in the memory set. This version of the election operator is weaker than that of Arifovic (1994) and similar to Franke (1997).

One could think of the three designs as implementing three sequential levels of reasoning. A trembling hand agent does not filter new strategies before putting them into practice. A Fresh score agent is a more thoughtful type and assigns a potential score to new strategies based on what would have been the outcome in the last period, so as to assess its potential performance this period. In addition to this evaluation, an agent endowed with the election operator compares the performance of old and new strategies to avoid discarding old strategies that are better than the new one.

In a social learning environment, there is no difference between the trembling hand and Fresh score design, because the effect of the Fresh score works through the choice rule (Figure 1. To understand the functioning of the Fresh score design in an individual learning GA, assume that there is no crossover but just mutation. Then in expectation, there are pK new strategies *available* in the memory set of each agent each period. What is the probability that a new strategy is *actually* played? In general, the probability is less than p (Propositions 3 and 4, below). Consider a situation with a continuous payoff function when all the strategies in the memory set are in a neighborhood of the equilibrium. A random new strategy is likely to rank lower here than in the memory set at period zero, which is randomly generated. In other words, the *actual* innovation rate, i.e. in terms of new strategies played, is going to change over time and become smaller when the GA agent is near equilibrium.

Proposition 3: Let X be the memory set, $X=X_A \cup X_B$, and X_A be the subset of old strategies and X_B be the subset of new strategies currently introduced by the innovation process.. If all available strategies have the same score, the probability that the chosen strategy x* is a new strategy, $P\{x^* \in X_B\}$,

a) is equivalent to the innovation level, p= $P\{x^* \in X_B\}$

b) is independentof the size of the memory set ♦

Proposition 4: If the sum of rankings of new available strategies within a memory set, $\sum_{x \in X_B} r_x$, declines, the probability that a new available strategy becomes the action, $P\{x^* \in X_B\}$, declines as well. ♦

Simulation result 3 (Election operator)

When the agents are endowed with an election operator, the individual heterogeneity level is lower than the level inthe basic GA agents' simulation only when the level of innovation is higher than 44%.♦

The results of the simulations under the trembling hand and election operator design are shown in columns (4) and (5), respectively, of Table 3. Not surprisingly, the trembling hand agents are more noisy at the aggregate and overall individual level than the Fresh score agents ($SD(X)_{TH}$=14.84 vs. $SD(X)_{FS}$=9.89 and $D1_{TH}$=18.80 vs. $D1_{FS}$=15.06). Trembling hand individuals are also slightly less heterogeneous ($D2_{TH}$=10.47 vs. $D2_{FS}$=11.08), due to a similar effect of the higher amount of noise in the Fresh score design, as explained in the discussion of Figure 2 Although it produces aggregate results closer to the Nash equilibrium outcome and with a dramatically reduced variance ($SD(X)_{EL}$=0.11 vs. $SD(X)_{FS}$=9.89), the election operator – surprisingly – does not decrease the amount of individual heterogeneity compared to the Fresh score design ($D2_{EL}$=11.91 vs. $D2_{FS}$=11.08). The surprise comes from the general view that the election operator characterizes agents with a higher level of rationality.

The election operator has a dramatic impact on the behavior of GA agents but does not lower individual heterogeneity. At the aggregate level, the result is similar to the work of Arifovic (1994) for the cobweb model. Without the election operator, there is a higher variability in the market's production (Figure 2B, which almost completely disappears with the election operator (Figure 3B). Yet, some other results are counterintuitive, as one would conjecture that a higher level of rationality, as the election operator is generally intended to induce, would lead to behavior that is closer to the symmetric Nash equilibrium at the individual level. To better investigate the functioning of the election operator, simulations were run varying the innovation rate as has been done with the baseline Fresh score GA design (Figure 2). The effect of the election operator on individual heterogeneity is not the same for all innovation rates. In comparison with the Fresh score design, a higher individual heterogeneity, according to both D2 and SD2 indexes, is detected for innovation rates between p=0.11 (pm=0.015) and p=0.44 (pm=0.070). For innovation rates above p=0.44, individual heterogeneity quickly declines below the level of the zero-intelligence agent (p=0.57, pm=0.1) and then toward zero; moreover, both individual heterogeneity and aggregate outcomes are closer to Nash equilibrium. Beyond p=0.81 (pm=0.185), the election operator seem to lose control of the inflow of new strategies from the high innovation rate, and the variance of aggregate outcome has a spike.

The innovation process is the counterbalance to the tendency to reinforce good strategies over time. In the Fresh score design, if the rate of innovation is too high there is a danger of corrupting the hard-learnt good strategies. With the election operator there is no such danger: when a new strategy does not promise to be better than its parent it does not get a chance of being played. As a matter of fact, it is immediately forgotten. In this context, the innovation rate needs a different interpretation than in the Fresh score design. One might think of it as an index of computational speed, i.e. of how many strategies the agent can create, evaluate, and compare in one period. The election operator with a high innovation rate induces a superior ability to explore currently unavailable options.

In a social learning GA all new strategies are automatically played and the election operator is the only way to filter out disruptive behavior (Figure 1). In an individual learning GA, instead, there is an additional filter between innovation and play, which is the choice rule. A new strategy with a low potential score has a low chance of being selected for play (Proposition 1) but will be kept in memory for future periods.

6 EFFECTS OF MEMORY CONSTRAINTS

Stronger memory capabilities make for a smarter decision maker. An agent with a larger memory size K has a longer historical memory (TOT) and abandons an available strategy only after a longer sequence of trials. Moreover, the decision maker has some advantages (either an advantage or some advantages) in the ability to choose a better strategy (Corollary 1).

Corollary 1: (i) The median ranking available strategy is chosen with probability 1/K.

(ii) The odds that the best versus the worst available strategy is chosen are increasing in the memory set size, (2K-1) (inverse of error odds).

(iii) Consider K even. The probability that the chosen strategy ranks above the median ranking available strategy is ¾, irrespective of the size of the memory set. ♦

While keeping the innovation level at p=0.15 (pm=0.02), we can study the effect of different memory sizes, letting K range from 2 to 100 (Figure 3). While augmenting noise (p) fades individual heterogeneity by generating overall variability (SD(X)), relaxing memory constraints makes for a better decision-maker with both lower individual heterogeneity (D2, SD2) and lower variance over time of individual actions (SD3). Numerical values for K=2 and K=90 can be found in columns (6) and (7) of Table 3. This result is also in line with the findings of the psychological literature: “differences in working memory capacity predict performance on a variety of tasks” (Daily et al., 2001).

A larger memory set systematically reduces individual heterogeneity. Initially the reduction in individual heterogeneity is fast and then it slows without stopping its decline. For K≥50 the genetic algorithm agents are always less individually heterogeneous than zero-intelligence agents and with a memory set as large as K=100 the individual heterogeneity is almost half that (D2=2.38 and SD2=0.80).[13] In conclusion, memory size matters (Result 4).

Simulation result 4 (Memory constraints)

When the rationality level of the agents is enhanced by enlarging memory capabilities, the individual heterogeneity level decreases toward zero. ♦

7 DISCUSSION AND CONCLUSIONS

In this paper we explore the effect of bounded rationality on the convergence of individual behavior to a given equilibrium. We show that constraints in terms of information processing capabilities and working memory can lead, in a game with a symmetric Nash equilibrium, to individually heterogeneous behavior. Moreover, as the rationality level increases, agents converge to uniform behavior.

Several experimental studies in economics report that under identical incentives people behave in a different fashion (Palfrey and Prisbrey, 1997; Saijo and Nakamura, 1995; Ledyard, 1995; Casari and Plott, forthcoming). One way to rationalize this evidence is to assume individual-specific utility functions. Alternatively, agents can have identical goals but differentiated skills. This study offers a third explanation: agents with identical goals and identical, although limited, levels of rationality.

Individual behavior is studied in the context of a Cournot game. In this game, fully rational agents should choose identical strategies. This paper presents simulated interactions of identical agents at several different levels of bounded rationality. The tool employed for agent-based modeling is an individual learning genetic algorithm (Holland and Miller, 1991; Vriend, 1998; Chen and Yeh, 2001). While allowing each agents to evolve based on its own experience, an individual learning genetic algorithm can be designed to fit many levels of agent rationality. Four analytical results regarding properties of genetic algorithm are presented in order to link the choice of the algorithm

[13] Consider that there are 256 possible strategy in the bynary coding.

design to behavioral assumptions (Propositions 1-4). In particular, the propositions concern the experimentation process, the effect of memory constraints, and invariance to payoff transformations.

In the baseline simulation, each agent can remember and process six strategies at a time, a number close to what is suggested about the working memory in the Psychology literature (Miller, 1956; Daily et al., 2001). Over time the best strategies gain a higher probability of being played. For each agent, new strategies are randomly generated (crossover and innovation) and introduced into her set of available strategies.

The simulations lead to four main results regarding the effect of bounded rationality on individual behavior. First, with limited information processing abilities and constrained working memory, individual actions are remarkably heterogeneous (Result 1). Second, even though one might suspect that the outcome is the result of the stochastic nature of some genetic algorithm operators (and hence it is built-in by construction), we show that this is not the case. Evidence from simulations indicates that genetic algorithm agents exhibit more individual heterogeneity than zero-intelligence agents (Gode and Sunder, 1993), who are essentially pure noise. In addition, lowering the innovation rate does not lead to homogeneous behavior. In other words, the heterogeneity result holds up besides the added noise (Result 2).

The other two results support the interpretation that individual heterogeneity is caused by bounded rationality. They indicate that as the level of agent rationality increases, individual heterogeneity fades away, yielding the standard prediction that fully rational agents have uniform behavior. Within the class of individual learning genetic algorithms, two dimensions of the bounds on rationality are explored: the innovation process and memory constraints. Relaxing memory constraints lowers individual heterogeneity, and the data suggest that it goes to zero for infinite memory capabilities (Result 4). This outcome of the computation model is in line with finding of the Psychology literature (Daily et al., 2001). In a separate set of simulations, the baseline model is adjusted by adding a more sophisticated evaluation of pre-play strategies in the form of an election operator. The individual heterogeneity does not disappear unless a high innovation rate is set (Result 3). In this context, innovation is interpreted as an index of computational speed of the agent.

To summarize, the contributions of this paper go into two directions: individual convergence to equilibrium and genetic algorithm design. First, it reports the existence of an inverse correlation between levels of rationality and levels of individual heterogeneity. In the limit, the simulations suggest that uniform behavior would result from full rationality. Second, it sheds light on the design of multi-population genetic algorithms. In particular, we explore the interaction of the election operator with the innovation rate and the working of the memory set in conjunction with a choice rule.

Interestingly, the simulation results also suggest that the Nash equilibrium is a more robust predictor of aggregate behavior than of individual behavior. It is as if a maximum range of individual diversity is compatible within a bound of agent capabilities. When there is a wide space of unexploited opportunities, even agents with heavy cognitive limits can find them and reap the gains but when the space is narrow they are not capable of doing so. As the search abilities rise along with rationality levels, the opportunities for gains disappear. For instance, if a firm grossly under-produces in a Cournot setting, there is an opportunity for another firm to “overproduce” and, as a result, market production could still be rather close to the aggregate Nash equilibrium outcome. However, more work is needed in this regard.

Other changes in the rationality level of the decision maker could be explored, such as the effect of a different rule to choose a strategy out of each of the individual sets of available strategies as well as an innovation process different from uniform binary mutation. They are all legitimate, and not mutually exclusive, possibilities to model the agents. This work is not a statement that any form of bounded rationality will lead to individual heterogeneity in behavior. In fact, in the context that we have analyzed only heavy bounds to rationality have produced it. The open issue is then how to calibrate these models to the actual cognitive limitations of people in order to understand if and how much of the individual heterogeneity observed in experimental data is due to bounded rationality.

APPENDIX: PROOF OF PROPOSITIONS

Proof of Proposition 1: Consider the ranking of available strategies x in A, {1,2, …, r_x, …, K} and a choice rule which operates by (1) drawing with replacement two available strategies out of A, and (2) taking the one with the highest score between the two. Let $p_x= p_a\cdot p_b$=P{x is drawn out of A}·P{x is chosen after it has been drawn}. There are three possible cases in which the available strategy x can be drawn, so the total probability is p_a=P{(x,y)}+ P{(y,x)}+ P{(x,x)}=$\frac{1}{K}\cdot\frac{K-1}{K}+\frac{K-1}{K}\cdot\frac{1}{K}+\frac{1}{K}\cdot\frac{1}{K}$, where y≠x. When the competing available strategy is y, the probability that available strategy x is chosen against any of the other K-1 is p_b=P{ $r_x> r_y$}=$\frac{r_x-1}{K-1}$. Hence, p_x=P{(x,y) or (y,x)}· P{$r_x> r_y$}+P{(x,x)}·1=$2\frac{1}{K}\left(\frac{K-1}{K}\right)\cdot\left(\frac{r_x-1}{K-1}\right)+\frac{1}{K^2}\cdot 1$.

The expression above defines a probability distribution since $\sum_{x\in MS} p_x=\sum_{r=1}^{K}\frac{2r-1}{K^2}=1$.♦

Proof of Corollary 1: (i) The median ranking is defined as $r_y=(K+1)/2$, hence $p_y=1/K$. (ii) P{r_x=1}=1/K^2, P{r_z=K}=(2K-1)/K^2, odds=p_z/p_x=(2K-1). (iii) Suppose K is an even number, $\alpha=\frac{\sum_{r=K/2+1}^{K}\frac{2r-1}{K^2}}{\sum_{r=1}^{K/2}\frac{2r-1}{K^2}}=3$, given that $\sum_{r=1}^{N} r=\frac{N(N+1)}{2}$ and $\sum_{r=w+1}^{K} r=w(K-w)+\sum_{r=1}^{K-w} r$.♦

Proof of Proposition 2: When the score s(a_{ik}) is replaced by v(s(a_{ik})), where v is a real function such that ∂v/∂·>0, the results of the decision process are not changed by the operations performed through the reinforcement rule, the innovation process, and the choice rule. The innovation process does not depend at all on the score. Both reinforcement and choice rules are based on a pairwise tournament, which operates on the ranking of the available strategies. As v does not change rankings, the results are unchanged. ♦

Proof of Proposition 3: Given an innovation level p (probability that an old available strategy is replaced by a new one), the expected number of new available strategies in a memory set of size K is E[|X_B|]=pK. When all available strategies have the same score, the probability that one of those new available strategies becomes an action is P{x*∈X_B}=(1/K) E[|X_B|]=p. ♦

Proof of Proposition 4: Consider the probability of an available strategy from the set X_B becoming an action,

P{x*∈X_B}= $\sum_{x\in X_B}\frac{2r_x-1}{K^2}=\frac{1}{K^2}\left(2R_B-pK\right)$, where $R_B=\sum_{x\in X_B} r_x$, and |X_B|=pK.

The proposition follows from ∂ P{x*∈X_B}/∂R_B>0. ♦

When new strategies are freshly evaluated, the expected number of new strategies that will be chosen is likely to decline as agents approach equilibrium. The actual ranking of a new available strategy a_{ikt} depends on how much learning has already taken place and on the nature of the innovation process itself.

REFERENCES

Andreoni, James and John H. Miller, 1995. Auction with Artificial Adaptive Agents. Games and Economic Behavior 58, 211-221.

Andrews, M. and Prager, R., 1994. Genetic programming for the acquisition of double auction market strategies. In: Kinnear, K.E. (Ed.) Advances in Genetic programming, Vol.1, MIT Press, 355-368.

Arifovic, Jasmina, 1994. Genetic algorithm learning and the cobweb model. Journal of Economic Dynamics and Control 18, 3-28.

Arifovic, Jasmina, 1996. The behavior of the exchange rate in the genetic algorithm and experimental economies. Journal of Political Economy 1104, 510-541.

Arifovic, Jasmina and John Ledyard, 2000. Computer Testbeds and Mechanism Design: Application to the Class of Groves-Ledyard Mechanisms for Provision of Public Goods. manuscript, Pasadena, California Institute of Technology.

Bäck, Thomas, 1996, Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms, New York, Oxford University Press.

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

Casari, Marco and Charles R. Plott, 2002. Decentralized Management of a Common Property Resource: Experiments with Centuries-Old Institutions. forthcoming, Journal of Economic Behavior and Organization

Chen, S.-H. and Yeh, C.-H., 1998. Genetic programming in the overlapping generation model: an illustration with dynamics of the inflation rate. In: Porto, V.W et al. (Eds.), Evolutionary Programming VII. Lecture Notes in Computer Science, Vol 1447., Berlin, Springer, 829-838

Chen, S.-H. and Yeh, C.-H., 2001. Evolving traders and the business school with genetic programming: A new architecture of the agent-based artificial stock market. Journal of Economic Dynamics & Control 25, 363-393

Daily, Larry Z., Lovett, Marsha C., Reder, Lynne M., 2001. Modeling individual differences in working memory performance: A source activation account. Cognitive Science 25, 3, 315-353

Dawid, Herbert, 1996, Adaptive Learning by Genetic Algorithms, Analytical Results and Applications to Economic Models, Springer, Berlin.

Dawid, Herbert, 1999, On the Convergence of Genetic Learning in a Double Auction Market. Journal of Economic Dynamics and Control 23, 1545-1569.

Ezekiel, M., 1938, The cobweb theorem. Quarterly Journal of Economics 52, 255-280.

Franke, Reiner, 1997. Behavioural Heterogeneity and Genetic Algorithm Learning in the Cobweb Model. IKSF - Institut für Konjunktur- und Strukturforschung,University of Bremen, no.9.

Gode, Dhananjay K., and Shyam Sunder, 1993. Allocative Efficiency of Markets with zero-Intelligence Traders: Market as Partial Substitute for Individual Rationality. Journal of Political Economy 1,101, 119-137

Goldberg, D., 1989, Genetic Algorithms in Search Optimization, and Machine Learning, New York, Addision-Wesley.

Hesser, J. and R. Männer, 1991. Towards an optimal mutation probability in genetic algorithms. in: Schwefel and Männer (Eds.) Parallel Problem Solving from Nature – Proceedings 1st Workshop PPSN I, val. 496 of Lecture Notes in Computer Science, Springer, Berlin, p. 23-32.

Holland, John H., 1975, Adaptation in Natural and Artificial Systems, Ann Arbor, University of Michigan Press.

Holland, John H. and John H. Miller, 1991. Artificial Adaptive Agents in Economic Theory. American Economic Review, Papers and Proceedings 81, 365-70.

Hommes, Cars H., 1998. On the consistency of backward-looking expectations: The case of the cobweb. Journal of Economic Behavior and Organization 33, 333-362.

Jensen, R.V and Urban, R., 1984, Chaotic price behavior in a nonlinear cobweb model. Economic Letters, 15, 235-240.

Kreps, David M., 1998. Bounded Rationality. entry of The new Palgrave dictionary of economics and the law, edited by Peter Newman, New York, NY, Stockton Press, 168-173.

LeBaron, Blake, 2000. Agent-based Computational Finance: Suggested Readings and Early Research. Journal of Economic Dynamics and Control 24, 679-702.

Ledyard, John O., 1995. Public Goods: A survey of Experimental Research, In: John H.Kagel and Alvin E.Roth (Eds), The Handbook of Experimental Economics 111-195.

Miller, G.A., 1956, The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. Psychological Review 101 (2), 343-352

Miller, John H. and James Andreoni, 1991. Can Evolutionary Dynamics Explain Free Riding Experiments? Economics Letters 36, 9-15.

Miller, John H., 1996. The coevolution of automata in the repeated prisoner's dilemma Journal of Economic Behavior and Organizations 29, 87-112.

Mitchell, Melanie , 1996. An Introduction to Genetic Algorithms. Cambridge, MIT Press.

Nowak, Martin A. and Sigmund Karl, 1998. Evolution of Indirect Reciprocity by Image Scoring. Nature 393, 573-576.

Palfrey, T.R., and Prisbey, J.E., 1997. Anomalous Behavior in Public Goods Experiments: How Much and Why?. American Economic Review 87, 5, 829-846.

Saijo, T., and Nakamura, H., 1995. The 'Spite' Dilemma in Voluntary Contribution Mechanism Experiments. Journal of Conflict Resolution 39, 3, 535-560.

Vriend, Nicolaas J., 2000, An Illustration of the Essential Difference between Individual and Social Learning, and its Consequences for Computational Analysis, Journal of Economic Dynamics & Control 24, 1-19.

Figure 1: INDIVIDUAL AND SOCIAL LEARNING GENETIC ALGORITHMS

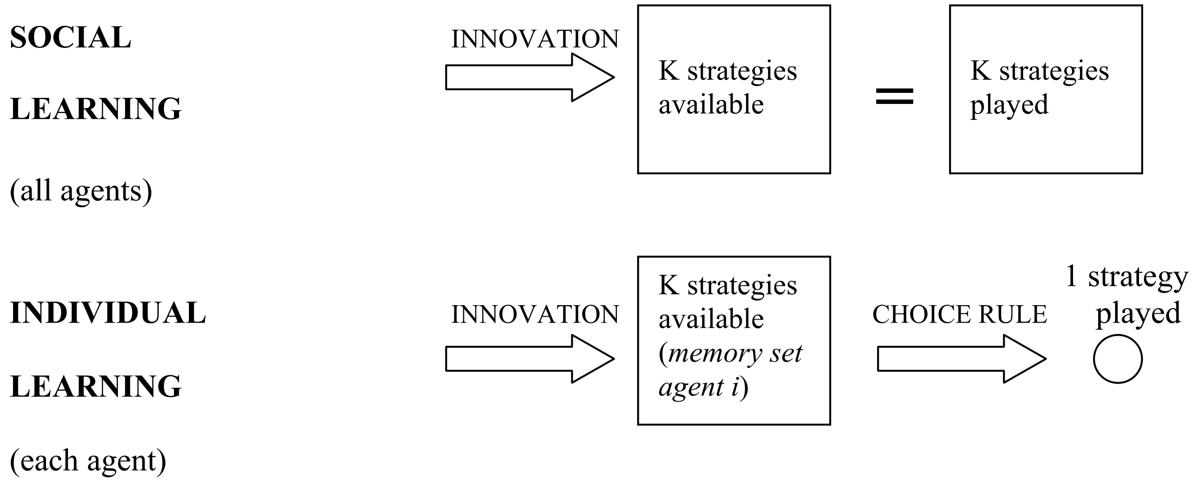
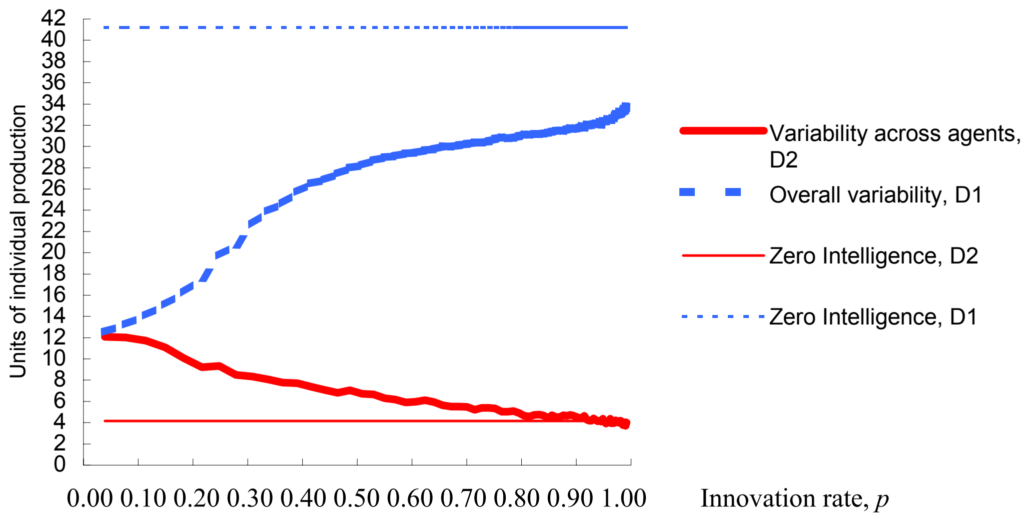
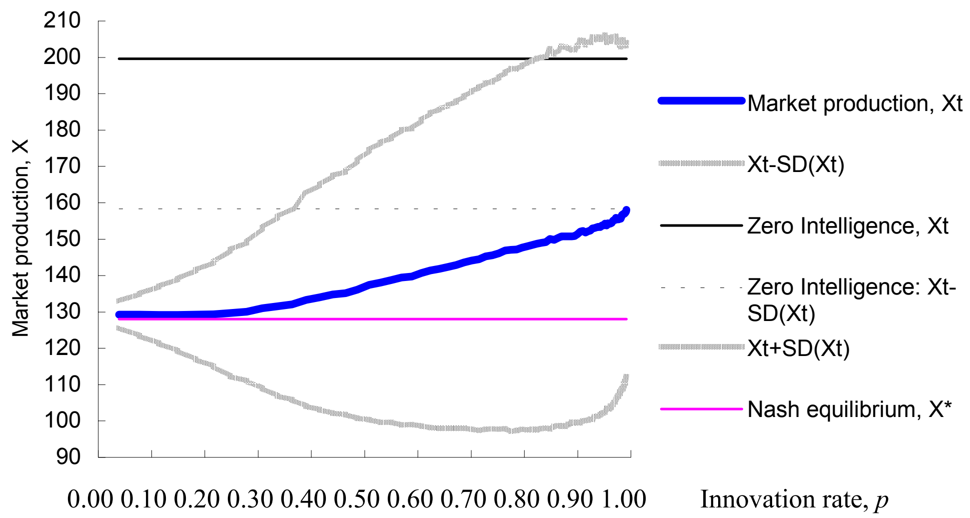


Figure 2: HETEROGENEITY AND INNOVATION RATE (FRESH SCORE)

A: Differences between maximum and minimum individual actions



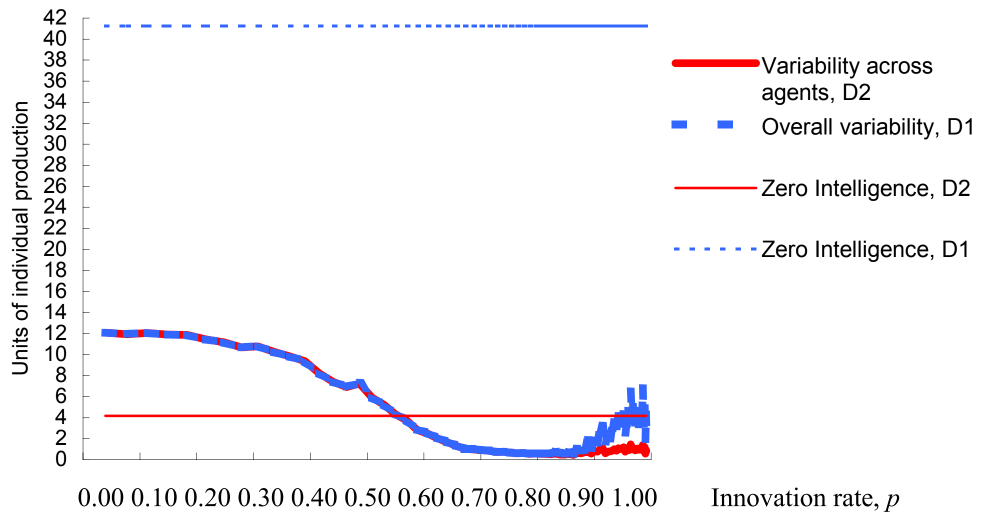
B: Market behavior



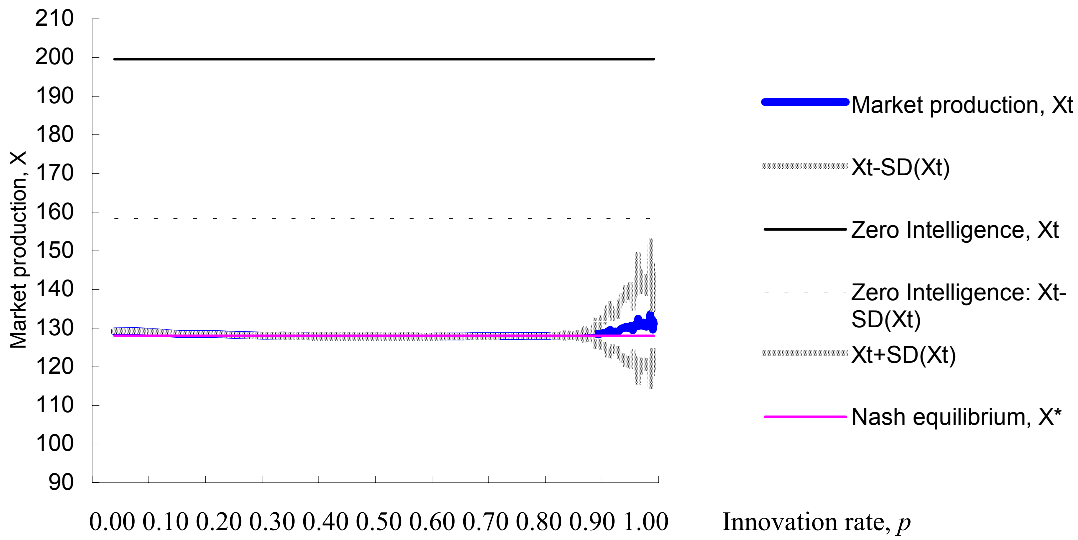
Notes: GA v.7.5, see Table 1; innovation rate $p=1-(1-pm)^L$, pm=mutation rate, L=string length; pm from 0.005 to 0.450; average of periods from 301 to 400 of 100 runs.

Figure 3: HETEROGENEITY AND INNOVATION RATE (ELECTION OPERATOR)

A: Differences between maximum and minimum individual actions



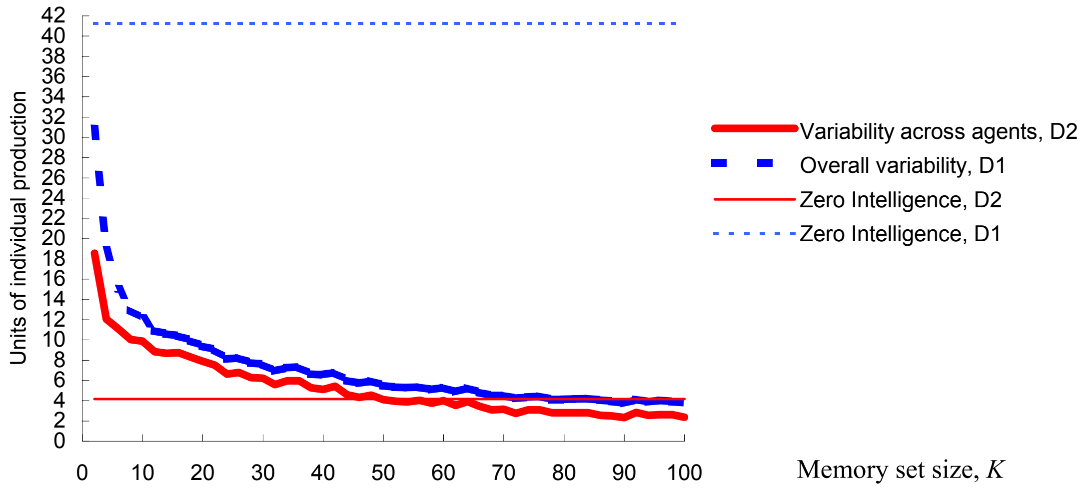
B: Market behavior



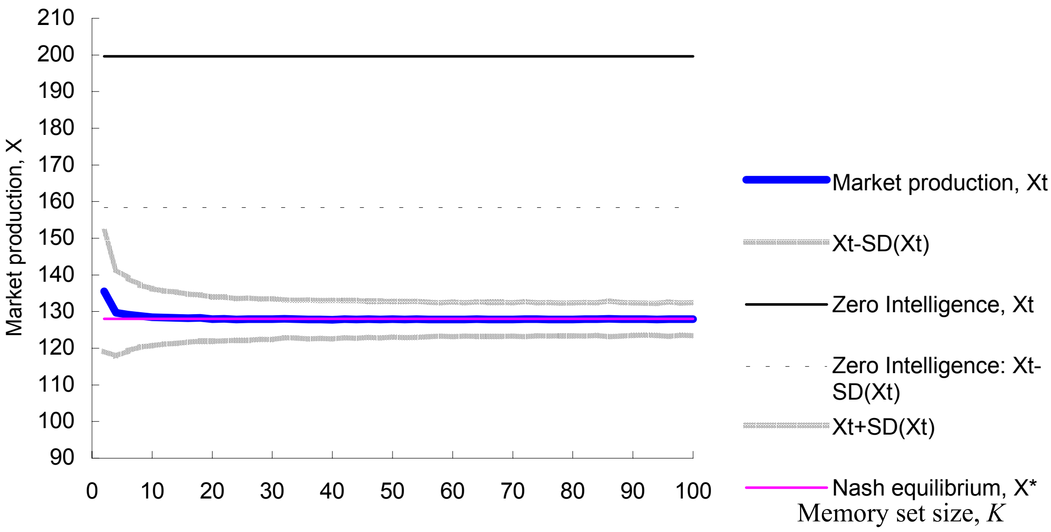
Notes: GA v.7.7 (Election), see Table 1; innovation rate $p=1-(1-pm)^L$, pm=mutation rate, L=string length; pm from 0.005 to 0.450; average of periods from 301 to 400 of 100 runs.

Figure 4: HETEROGENEITY AND MEMORY CONSTRAINTS

A: Differences between maximum and minimum individual actions



B: Market behavior



Notes: GA v.7.5 (Fresh score), see Table 1; K from 2 to 100 at interval of 2; average of periods from 301 to 400 of 100 runs.

Table 1: THE DESIGN OF THE GENETIC ALGORITHM

Number of agents, N	8
Number of strategies for each agent, K	6
Length of binary string, L	8
Range in decimal values	0-255
Probability of mutation, pm	0.02
Probability of crossover, pc	0.30
Crossover type	Single cut
Reinforcement rule (selection)	Pairwise tournament
Choice rule	Pairwise tournament
Initialization of strategies	Random from uniform distribution
Number of runs	100 different random seeds

Note: The GA agents were programmed and the simulations run on a PC using Turbo Pascal.

Table 2: EXAMPLES OF TWO PATTERNS OF INDIVIDUAL VARIABILITY

Scenario	Agent	Period				*Agent average* $\overline{x_i}$	Indexes of variability of individual actions				
		1	2	3	4		Overall D1	Overall SD1	Across agents D2	Across agents SD2	Over time SD3
A	x_1	12	12	12	12	*12*	10	5.35	10	7.07	0
	x_2	22	22	22	22	*22*					
B	x_1	12	22	12	22	*17*	10	5.35	0	0	5.77
	x_2	22	12	22	12	*17*					

Note: D=difference between maximum and minimum, SD=standard deviation

Table 3: SIMULATION RESULTS

	(1) Nash equilibrium	**(2) Fresh score K=6 p=0.15**	(3) Fresh score K=6 p=0.01	(4) Trembling hand K=6 p=0.15	(5) Election K=6 p=0.15	(6) Fresh score K=2 p=0.15	(7) Fresh score K=90 p=0.15	(8) Zero Intelligence agents
MARKET RESULTS								
Production	*128*	**129.18**	129.25	130.54	128.55	135.49	127.92	199.58
Standard deviation of production	*0*	**9.89**	1.79	14.84	0.11	16.30	4.43	41.23
Price	*3.5*	**3.41**	3.42	3.34	3.47	3.01	3.48	-0.98
Standard deviation of price	*0*	**0.65**	0.13	0.93	0.01	1.03	0.37	2.56
Profits (% of monopoly profits)	*39.5%*	**34.97%**	36.60%	29.59%	38.29%	16.48%	39.27%	-246.95%
INDIVIDUAL AGENT RESULTS *(1 obs= production decision for one agent at time t)*								
MIN1 – Minimum production across agents *(average across runs and periods)*	*16*	**9.93**	10.40	8.18	12.33	3.68	13.93	5.44
MAX1 – Maximum production across agents	*16*	**24.99**	24.63	26.98	24.24	34.52	17.65	44.53
D1 – Difference	*0*	**15.06**	14.24	18.80	11.91	30.84	3.72	39.09
INDEXES OF INDIVIDUAL HETEROGENEITY *(1 obs= average production for the same agent over τ periods)*								
MIN2 – Minimum production across agents (average across runs)	*16*	**11.73**	10.31	11.85	12.33	8.66	14.51	22.82
MAX2 – Maximum production across agents	*16*	**22.81**	24.71	22.31	24.24	27.23	16.85	26.99
D2 – Difference	*0*	**11.08**	14.40	10.47	11.91	18.57	2.35	4.17
SD2 – Standard deviations of individual production	*0*	**3.68**	5.28	3.48	4.43	6.57	0.78	1.41
INDEX OF INDIVIDUAL VARIABILITY OVER TIME *(1 obs = sd for one agent over τ periods)*								
SD3 – Standard deviations of individual production	*0*	**3.51**	0.38	5.46	0.04	8.16	1.31	14.51

Notes: The statistics are computed on periods 301-400 and are averages over 100 runs with different random seeds 0.005-0.995. (2) Genetic Algorithm v.7.5 (Fresh Score), N=8, L=8, T=400, τ=100, K=6, pc=0.30, pm=0.02; (3) same as (2) with pm=0.001256; (4) Genetic Algorithm v.7.6 (Trembling hand), N=8, L=8, T=400, τ=100, K=6, pc=0.30, pm=0.02; (5) Genetic Algorithm v.7.7 (Election), N=8, L=8, T=400, τ=100, K=6, pc=0.30, pm=0.02; (6) same as (2) with K=2; (7) same as (2) with K=90; (8) Genetic Algorithm v.7.5.1 (zero intelligence), N=8, L=8, individual actions are drawn with replacement from an uniform distribution on [0,50].