DIVISION OF THE HUMANITIES AND SOCIAL SCIENCES
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA 91125

Learning to Alternate

John Ledyard
California Institute of Technology

Jasmina Arifovic
Simon Fraser University

SOCIAL SCIENCE WORKING PAPER 1437
February 2018

Experimental Economics manuscript No.
(will be inserted by the editor)

Learning to Alternate

Jasmina Arifovic · John Ledyard

February 21, 2018

Abstract The Individual Evolutionary Learning (IEL) model explains human subjects' behavior in a wide range of repeated games which have unique Nash equilibria. Using a variation of 'better response' strategies, IEL agents quickly learn to play Nash equilibrium strategies and their dynamic behavior is like that of humans subjects. In this paper we study whether IEL can also explain behavior in games with gains from coordination. We focus on the simplest such game: the 2 person repeated Battle of Sexes game. In laboratory experiments, two patterns of behavior often emerge: players either converge rapidly to one of the stage game Nash equilibria and stay there or learn to coordinate their actions and alternate between the two Nash equilibria every other round. We show that IEL explains this behavior if the human subjects are truly in the dark and do not know or believe they know their opponent's payoffs. To explain the behavior when agents are not in the dark, we need to modify the basic IEL model and allow some agents to begin with a good idea about how to play. We show that if the proportion of inspired agents with good ideas is chosen judiciously, the behavior of IEL agents looks remarkably similar to that of human subjects in laboratory experiments.

Keywords: Battle of Sexes, Alternation, Learning

JEL classification: C72, C73, D83

We thank Kevin James, Brian Merlob and Heng Sok for their excellent research assistance. We would also like to thank John Duffy, Tim Cason, Julian Romero, participants at the Workshop in Memory of John van Huyck, Southern Methodist University, 2015, participants at the Southern Economic Association Meetings, New Orleans, 2015, as well as two referees and an editor. Jasmina Arifovic gratefully acknowledges financial support from CIGI-INET grant # 5553. John Ledyard thanks the Moore Foundation whose grant to Caltech for Experimentation with Large, Diverse and Interconnected Socio-Economic Systems, award #1158, supported the experimental work.

J. Arifovic
Simon Fraser University
E-mail: arifovic@sfu.ca

J.Ledyard
California Institute of Technology
E-mail: jledyard@caltech.edu

1 Introduction

We are interested in constructing a theoretical model of the behavior of human subjects in finite repeated games in laboratory settings. We have already studied repeated versions of call markets, voluntary contributions, and Groves-Ledyard mechanisms, and have shown that Individual Evolutionary Learning (IEL) explains the data from experiments (Arifovic and Ledyard, 2007, 2010, 2011).[1] Each of those games has a well-defined, unique stage game Nash-equilibrium, and little gain from coordinated play. As a result, both human subjects and IEL agents converge rapidly to the Nash equilibrium. But there are other interesting games where coordination has advantages. One canonical example is the Battle of Sexes. In this paper we explore how well IEL explains data from Battle of Sexes experiments.

Our research objectives and methodology have a lot in common with John Van Huyck's research. He did pioneering work in the investigation of coordination and equilibrium selection in games with a multiplicity of equilibria. Together with his co-authors, he investigated these issues conducting experiments with human subjects and comparing the predictions of various adaptive algorithms with experimental outcomes.[2] A good example can be found in Van Huyck et al. (1994) where they examine human behavior in a coordination game with multiple equilibria in which the myopic best-response dynamic and inertial selection dynamic make different predictions about stability. Both of these dynamics belong to the class of 'relaxation algorithms'.[3] They find that the inertial selection dynamic accurately predicts the behavior observed in the experiment while myopic best-response dynamic does not. We build upon this line of research, and develop a behavioral model that is intended to capture 'real time' dynamics that characterize experiments with human subjects.

In experiments with Battle of Sexes games, inter-temporal patterns of play emerge quickly. Subjects often learn to alternate between the two pure strategy, stage game equilibrium outcomes. See, e.g., Rapoport, Guyer and Gordon (1975) and McKelvey and Palfrey (2002). Other pairs converge to playing the stage game Nash Equilibrium. Sonsino and Sirota (2003) conduct repeated asymmetric Battle of Sexes experiments and analyze the results from the perspective of "strategic pattern recognition." They show that human subjects are able to sustain non-trivial alternating patterns of the stage game Nash equilibria with more than half of the pairs of subjects weakly converging to a fixed equilibrium pattern. The percentage of pairs that alternate and the percentage that converge to a Nash Equilibrium vary considerably across different payoff matrices and different information conditions.

As we discuss more fully in Section 1.2, the challenge is to provide a theoretical explanation of these patterns of behavior across all treatments: symmetric and asymmetric. One way would be to assume fully strategic behavior and common knowledge of rationality, and then show that playing a stage game Nash equilibrium and playing an alternating plan are two of the many equilibria of the game. But there are also many more such equilibria, so this is not particularly explanatory.

Instead we begin by considering pure reactive learning. In particular, we ask whether IEL will produce behavior similar to that in the experiments. In IEL, agents are not given any hints. The model does not

[1] For those unfamiliar with IEL, we provide a non-technical description in Section 2.2.1 and a complete technical description in Section 6 of the Appendix.

[2] See, for example, Van Huyck et al. (1990); Van Huyck et al., (1991); Van Huyck et al. (1997); Van Huyck et al. (2007a); and Van Huyck et al. (2007b).

[3] Basically, it is any adjustment algorithm that describes out-of-equilibrium adaptation of the following form: $X_k^* = X_{k-1}^* + \lambda(X_k^* - X*_{k-1})$ where $\lambda \in [0, 1]$ is a so-called 'relaxation parameter' and X_k^* is the estimate of the of the equilibrium value of an economic variable at the k^{th} iteration. See Sargent (1993) for a definition and a detailed description of the 'relaxation' algorithm.

involve calibration. Agents begin with a purely random collection of possible plans. As play progresses they modify this collection, keeping plans that would have been paid well and replacing plans that would not have been paid well. We show that this model explains the behavior of human agents who are truly in the dark when they play the repeated game: that is, they do not know or think they know what their opponent's payoff matrix is. But we have to modify the IEL model slightly to explain the behavior of those who do know their opponent's payoffs.

Before we explain that modification and analyze the data, we introduce the repeated Battle of Sexes game and discuss some of the theoretical literature. We then provide a guide to the paper which is also a brief summary of what we did and what we found.

1.1 The Repeated Battle of Sexes Game with a Strategy Continuum

1.1.1 The Traditional One-shot Game

The traditional 2-person, 2-action Battle of the Sexes game has the strategy sets $A^i = \{0, 1\}$ and the payoff functions $u^i(1, 1) = \alpha^i$, $u^i(1, 0) = \gamma^i$, $u^i(0, 1) = \beta^i$, $u^i(0, 0) = \delta^i$ where $\beta^1 > \gamma^1 > \alpha^1 = \delta^1$ and $\gamma^2 > \beta^2 > \alpha^2 = \delta^2$. A symmetric example of such a stage game has $\beta^1 = \gamma^2 = 15, \beta^2 = \gamma^1 = 9$, and $\alpha^1 = \delta^1 = \alpha^2 = \delta^2 = 0$. The payoff matrix is found in Table 1.

	1	0
1	0, 0	9, 15
0	15, 9	0, 0

Table 1: Symmetric BoS
payoff matrix

There are 3 Nash Equilibria in this game; 2 pure and 1 mixed. The pure equilibria are $(1, 0)$ and $(0, 1)$. The mixed equilibrium is $(3/8, 3/8)$.

1.1.2 The Traditional Repeated Game

In a repeated game, a stage game (A, u) is played for T rounds. In the stage game, $\mathcal{A} = A^1 \times ... \times A^N$ where $A^i = A^i_1 \times ... \times A^i_t \times ... \times A^i_T$ are the actions available to i, with A^i_t being the actions available in round t, and $U = (U^1, ..., U^N)$ where $U^i = \sum_{t=1}^{T} u^i(a_t)$ with $u^i : \mathcal{A} \to \Re$ being i's stage game payoff function.

In a repeated game, a strategy can be more complex than simple alternation. A history of play at round t is $h_t \in \mathcal{H}_t = \mathcal{A}^1 \times ... \times \mathcal{A}^{t-1}$. A strategy for i is $s^i = (s^i_1, ..., s^i_T)$ where $s^i_t : \mathcal{H}_t \to \Delta(A^i)$, where $\Delta(A)$ is the set of probability measures with finite support on A. There are many equilibria of the repeated game. For example, consider the Pareto set, $\mathcal{P} = \{(\bar{u}^1, \bar{u}^2) | \bar{u}^i = \lambda u^i(1, 0) + (1 - \lambda) u^i(0, 1), \lambda \in [0, 1]\}$. In a finitely repeated game, the Pareto set is partially attainable on average with alternating strategies of various lengths. For example, if the two players were to use $(0, 1)$ for k rounds and then switch to $(1, 0)$ for j rounds and then switch back to $(0, 1)$ and so forth, the average payoffs would be $(\frac{15k+9j}{k+j}, \frac{9k+15j}{k+j})$. Here $\lambda = \frac{k}{k+j}$. By suitably choosing k and j one can get many of the rational payoffs in $\mathcal{P}$.

1.1.3 The Generalized Repeated Battle of Sexes Game

We generalize the 2-action games by considering a continuum of stage game actions with $A_t^i = [0, 1]$. We do this for the following reason. Although mixed strategies are not particularly salient or expected in repeated Battle of the Sexes games, we are looking for a way to have subjects consider something like them and their certainty equivalent.[4]

We use payoff functions $u^1(a_1, a_2)$ and $u^2(a_1, a_2)$ such that the pure strategy Nash equilibria of the 2-action stage games are preserved. These are

$$u^i(a_1, a_2) = \alpha^i + (\gamma^i - \alpha^i)a_1 + (\beta^i - \alpha^i)a_2 + (\alpha^i - \beta^i - \gamma^i + \delta^i)a_1a_2 \tag{1}$$

For the payoff matrix in Table 1, this yields

$$u^1(a) = 9a_1 + 15a_2 - 24a_1a_2$$

$$u^2(a) = 15a_1 + 9a_2 - 24a_1a_2$$

It is important thing to note is that the action space for our generalized repeated Battle of the Sexes game does NOT include mixed strategies but DOES include real numbers.

If $a_t^1 = 0.4$ were a mixed strategy for player 1 and $a_t^2 = 0$, this would mean that we would roll a ten-sided die and let 1's action be 1 if the die shows 1-4 pips and 0 otherwise. Player 1's action would be either 0 or 1 and player 1 would get a payoff of either 9 or 0. The payoff to player 1 would therefore be 9 with probability 0.4 and 0 with probability 0.6. The *expected* payoff to player 1 would be 2.25. This is not what we do.

In our generalization of the Battle of thee Sexes games, subjects can play any real number between 0 and 1. They are not restricted to playing only integers. An action a_t^i such that $0 < a_t^i < 1$ is <u>not</u> a mixed strategy. Choosing $a_t^i = 0.3$ does not mean that i plays 1 with probability 0.3 and plays 0 with probability 0.7. Rather it means i plays 0.3 with certainty. We have created payoff functions for these real number actions which are the certainty equivalent, for a risk-neutral player, if it were a mixed strategy. In the experiments and the simulations below, when player 1 chooses $a_t^1 = 0.4$ and 2 chooses $a_t^2 = 0$, we pay player 1 the amount of 2.25 and player 2 the amount of 3.75.

Played repeatedly, the Battle of Sexes is a simple, canonical game in which there is a tension between individually best actions (stage game Nash Equilbria) and coordination (alternating plans that share the payoffs). We are interested in the strategies that human subjects use in laboratory experiments in repeated Battle of Sexes games, and how consistent our IEL behavioral model is with the behavior of those human subjects.

1.2 Literature

Standard learning theories ignore information about the game and do not capture features observed in the experiments. The theoretical literature on repeated games started with the use of fictitious play (Brown, 1951) where in each round, each player best responds to the average of past plays.

[4] Our preliminary simulation results as well as pilot experimental sessions indicate that the main results in this paper would also be true if $A_t^i = \{0, 1\}$.

Another approach was to use Cournot Best Reply dynamics.[5] A third approach is based on a notion of reinforcement, see Bush and Mosteller (1951). The modern incarnations of this can be found in the Reinforcement Learning (RL) model of Erev and Roth (1998) and the Experience Weighted Attraction (EWA) of Camerer and Ho (1999). In Battle of Sexes games, these models generate behavior that converges to a stage game Nash Equilibrium - alternation does not arise.

Myung and Romero (2013) develop a pattern recognition algorithm to study a continuous version of two repeated coordination games: minimum effort and battle of sexes. Agents have a specific mechanism to recognize patterns of different lengths and select the optimal pattern length. Out of 300 simulations that they conduct for the Battle of Sexes game, only 0.5% of the simulations converge to alternation - significantly less than occurs in experiments with humans.

Sophisticated learning algorithms based on social learning produce alternation, but seem unrealistic; requiring thousands of learning periods. These models do two things that standard learning models do not: they expand the space over what the agents are learning and they introduce a 'pre-experimental' phase. Hanaki et. al. (2005) and Ioannou and Romero (2014) both consider automata, Moore machines, as the basic strategy space. This leads to 26 possible strategies. Of course this complexity requires a lot of learning. Hanaki et al. (2005) summarize the problems related to the development of strategic learning models. These difficulties include the complexity of the strategy spaces and the complexity of the computation of counterfactual payoffs required in the updating of the learning algorithms.

Both papers consider only symmetric, full information games. Both papers introduce a 'pre-experimental' phase to accomplish the necessary learning. In Hanaki et. al. (2005), in that phase players use the same plan for several periods, and occasionally switch as part of the experimentation process. This can take a long time. In Ioannou and Romero (2014), the 'pre-experimental' phase consists of a lot of 100 period epochs without rematching. These epochs continue until average payoffs are stable for at least 20 epochs. This usually requires thousands of epochs.

In the 'experimental phase,' in both papers there is a fixed number of periods without further rematching. A learning process occurs during this stage as well, building on the attractions generated during the pre-experimental phase. For Hanaki et. al. (2005), the duration of this phase is 50 periods which corresponds to the duration of McKelvey and Palfrey's experiments. Agents use a Reinforcement Learning type of algorithm. But, because all agents have been through the 'pre-experimental' phase and most have learned to alternate, there is too much alternation relative to the experimental data. In Ioannou and Romero (2014), in the 'experiment' phase everyone is randomly rematched at the beginning and there is a 100 period game. They use three different learning models during this stage: a self-tuning Experience Weighted Attraction model (Ho et al., 2007), a γ -Weighted Beliefs model (Cheung and Friedman, 1997), and an Inertia, Sampling and Weighting model (Erev et al., 2010).

What is needed is a theory that does not require extensive training and homogenization of the subjects in a "pre-experimental phase". Best would be if no training were required. IEL is such a model.

1.3 A Guide to the Paper

We take a little detour getting to our final results. We do so because we think this leads to a deeper understanding of the experiments we run, the IEL learning model, and exactly how they relate to

[5] See Boylan and El Gamal (1993), for an experimental evaluation of these models.

each other. For those who want to get right to the results, we suggest they first read Sections 2.1.1 and 2.2, to get the basics, and then go to Sections 3 to 5. For everyone else, we provide a brief guide to the paper.

We begin in Section 2 by reporting on results from applying our usual methods to the Battle of Sexes game.[6] We run some experiments with human subjects, run some simulations with IEL agents, and compare the outcomes that occur. The experiments we run are the natural ones where subjects know the payoff matrix and are told the action of their opponent after each round. We call that ‘Full’ information. The simulations used IEL, modified for multi-period plans of length up to 4 periods. When we do this we find that IEL agents do not alternate as much as human agents do. That is, IEL does not explain the human behavior in those experiments.

But IEL operates in a different environment than that of the experiments. An IEL agent only knows their own part of the payoff matrix and the action of the others after each round. This is, in the terminology of McKelvey and Palfrey (2002), ‘Playing in the Dark’.[7] We call this the Dark information environment. There are two options at this point. Change the human experiments to correspond to the Dark information environment or change IEL to match the human behavior in Full information environments. We take these up in order.

In Section 3, we modify the human experiments to match the the ‘Dark’ environment of IEL. To augment that environment in the laboratory, we also introduced an asymmetric payoff matrix. Comparing the outcomes of experiments with the Dark information and asymmetric payoffs to the outcomes of IEL simulations with multi-period plans of length no greater than 2, we get virtually identical percentages of behavior that converge to a stage game Nash Equilibrium and that converge to Alternation. That is, IEL does explain the human behavior in those experiments.

In Section 4, we modify IEL in a minimal way to match human behavior with Full information. We allow some IEL agents to have an initial inspiration that alternating might be a good plan. This is equivalent to adding one parameter to the model. This inspiration might come from social learning in prior experiences with turn taking or from strategic analysis of the game. We collect data from experiments in all four situations: Full information and symmetric payoffs, Dark information and symmetric payoffs, Full information and asymmetric payoffs, and Dark information and asymmetric payoffs. We then solve for the percentage of IEL agents with inspiration that best explains the experimental data. For the Dark environments, IEL with inspiration explains the human data very well. For the Full information environments, IEL with inspiration goes a good ways towards explaining human data, but not as well as for the Dark. We suggest a possible reason for that.

We summarize the results of the paper in Section 6 and provide a few additional thoughts.

2 Our Usual Method

In our past work with IEL, our usual approach was to choose the game situation we wanted to study, run some laboratory experiments for that game, run some simulations with IEL for that game, and then compare the outcomes that occurred. We begin with that same approach here.

[6] For earlier examples of this type of analysis see Arifovic and Ledyard (2007), (2011), and (2012).

[7] McKelvey and Palfrey (2002) used this terminology to refer to their set of experiments with human subjects where subjects did not know the payoff matrix of players they were matched with.

2.1 Experiments

In this section we report on the initial experiments we ran with repeated Battle of Sexes games.

2.1.1 The Experimental Design

The subjects were recruited through the Caltech Social Science Experimental Laboratory. For reasons that will become apparent below, we tried very hard to recruit only those with limited experience in games where turn taking might be important. 20 pairs participated. Each pair participated in only one match for 40 rounds and this match occurred at the end of a market experiment.

We used the payoff matrix in Table 1 with a continuous action space $S = [0, 1]$. In each round, each subject chose a number $a \in S$.[8] We used the zTree experiment software package (Fischbacher, 2007). Subjects were given a paper copy of their payoff table that showed their payoffs in the increments of 0.1 for each combination of their action and the action of the player with which they were matched. We also provided subjects with a 'what-if-calculator' on their zTree screen that they could use to compute their payoff and their opponent's payoff for any combination of their action and their opponent's action.

2.1.2 Results

In analyzing the results of the experiment, we are interested in those pairs of sequences of actions subjects learn to coordinate on over the 40 rounds. We focus on two that are equilbria in the repeated game.

– *Nash equilibrium*: One obvious joint sequence of actions is for each player to repeatedly play their component of a pure plan Nash equilibrium in the stage game. That is, the joint play is either (0,1) or (1,0) every round.
– *Alternating equilibrium*: Another fairly obvious joint sequence of actions is for the two players to alternate between (0,1) and (1,0) every other period. In some sense, the payoff from this sequence is fair. Further, there is little risk to either player in this coordination. Such alternation is also consistent with what has been found in experiments with Battle of Sexes.

In Figure 1, we provide an example of each from the experimental data of each of these joint sequences. We also exhibit an alternating sequence that has length longer than 2 and one that is unclassifiable. In the rest of this paper, we call such plans Other.

For each pair of subjects in each experiment, we classify the collection sequences of actions into those that converge to a Nash equilibrium and those that converge to an Alternating equilibrium. To do so we look at the last four moves (actions) that two players played. Let a^1_{T-3}, a^1_{T-2} , a^1_{T-1}, and a^1_T denote the last four moves of player 1, and a^2_{T-3}, a^2_{T-2} , a^2_{T-1}, and a^2_T denote the last four moves of player 2. Then, we compute the following 3 measures that we use to determine what the player i's collection of plans is converging to:

$m_1 = (a^1_{T-1} - a^1_T)^2 + (a^1_{T-3} - a^1_{T-2})^2$ to measure if player 1 alternates during the last 4 moves.

[8] In fact, subjects could only choose in increments of 0.01.

$m_2 = (a^2_{T-1} - a^2_T)^2 + (a^2_{T-3} - a^2_{T-2})^2$ to measure if player 2 alternates during the last 4 moves.

$m_3 = (a^1_T - a^2_T)^2 + (a^1_{T-1} - a^2_{T-1})^2 + (a^1_{T-2} - a^2 T - 2)^2 + (a^1_{T-3} - a^2_{T-3})^2$ to see if player 1 and player 2 play different actions in each of the last 4 rounds.

If players have converged to a stage game Nash equilibrium, then $m_1 = 0$, $m_2 = 0$ and $m_3 = 4$. We use the following conditions to classify a result of the run as converging to Nash equilibrium: $m_1 + m_2 < 1$, and $m_3 > 3.5$. If players alternate in the last four moves, then $m_1 = m_2 = 2$ and $m_3 = 4$. We require that $m_1 + m_2 > 3$ and that $m_3 > 3.5$ to classify a result of a run as converging to Alternation. If neither of the conditions for Nash equilibrium or Alternating equilibrium are satisfied, a run is classified as 'Other '.[9]

It should be noted that this scheme of classification can lead to the inclusion as Nash Equilibria of some sequences with alternation. For example, if the subjects are alternating every 5 periods, then $m_1 + m_2 = 0$ and $m_3 = 4$ even though it is not a Nash Equilibrium. Something like this occurred in 16% of the experiments that we ran. We do not believe this has any adverse consequence for our conclusions in this paper.[10]

We summarize the results of the experiment in Table 2. The column labeled "%" is the percent of pairs in that treatment that ultimately played that particular joint strategy. The column labeled "period started" indicates the average of the period at which a pair locked onto this joint strategy.

plan	%	period started
Nash Equilibrium	30	5.5
Alternate	40	8.25
Other	30	na

Table 2: Results of BoS experiment

We find, as do others, some Alternating, some Nash equilibrium play and some Other. The latter could be due to confusion or disagreement. The starting times are instructive. Those pairs who settle on either Nash or alternate do so fairly quickly. Alternation takes a little longer. We defer further comment until later.

2.2 Simulations

In this section, we report on a set of simulations run with a repeated Battle of Sexes game using IEL agents. The IEL model is reasonably straight-forward. We provide a brief non-technical description here for those who are unfamiliar with it. A formal description of IEL is provided in the Appendix in Section 6.

[9] Only 10% of the pairs, from all of the 80 experiments conducted for this paper, did not converge to integer values for m_1, m_2 and m_3. In all cases, $m_3 < 3.5$ and so we classified these as Other.

[10] If we were to reclassify those as Alternate and recompute everything below, it would change the quantitative calculations a little, but it would not affect the qualitative conclusions.

2.2.1 IEL agents

IEL is based on an evolutionary process which is individual, and not social. IEL is particularly well-suited to repeated games with large action spaces such as convex subsets of multi-dimensional Euclidean space. At the heart of each IEL agent's strategy is a finite set of possible plans of action that they carry in their memory from round to round.[11] A plan is simply a sequence of actions, such as (1, 0, .5, .3, 0), and can be of any length. Plans are evaluated by looking backward and asking what would the average per period payoff have been if that plan had been used and others took the actions they did. We call that the foregone utility of the plan.

When an IEL agent chooses their action in a round they do so by either playing the next entry in a plan they are currently using or randomly selecting one of their considered plans in proportion to its foregone utility and then playing the first action in that selected vector. This is a particularly simple selection process but the particular selection rule does not matter much. The set of considered plans becomes homogeneous very quickly, and once approximate homogeneity is attained, selection becomes irrelevant.

An IEL agent updates their considered set of plans from period to period in an evolutionary manner. This occurs in three steps. The first step involves *rotation*. Each plan is moved forward one time period; that is, the first entry now becomes the last, the second the first, etc. The second step involves a little *experimentation* so that the agent will consider additional plans other than those currently under consideration and will not get stuck on non-optimal actions. Each plan in the agent's considered set is changed with some small probability. The experimentation can be in length (make it longer or shorter) or in value (each action can be increased or decreased). The third step involves *replication* to get rid of considered plans that would not have paid well - those with low foregone utility. In replication, a new set of considered plans is created by drawing 2 plans at random from the old set with replacement and keeping the one with the higher foregone utility. Doing replication after experimentation means that those modified plans that would not have done well in the past are quickly dispensed with and neither the agent nor their opponents are led in an unprofitable direction because of experimentation.

In each round, the agent is either continuing to follow a multi-period plan or choosing a new plan. In choosing a new plan the agent uses a mixed strategy over their considered set with probabilities proportional to the foregone utility of the plan - what it would have earned if it had been used against the actual past actions of the opponents. If the considered set is homogeneous, this just picks that single plan. So the crucial part of IEL is the updating process in which the considered set of an agent co-evolves with the other agents' sets. As the repeated game proceeds through rounds, experimentation and replication lead to an agent's considered set becoming more homogeneous, with lots of duplicates that are best replies to the plans the other agent is using.

If all agents are IEL agents, then the learning process tends to converge close to a Nash Equilibrium. Not a Nash Equilibrium in the stage game but a Nash Equilibrium in the space of considered plans. If there is a unique Nash Equilibrium in the space of plans that is the end of the story. On the other hand if there are multiple equilibria, initial conditions may matter. As we will see, that is what happens in the repeated Battle of Sexes game.

As mentioned above, we provide a detailed, technical description of IEL in the Appendix, including the values of the parameters we used in this paper. In addition, in the Supplemental Material, we

[11] This set can and will contain duplicates.

present a toy example of how an IEL simulation proceeds as well as our Python code that we used for our IEL simulations.

2.2.2 Results

For the simulations with IEL, we used the same payoff matrix as in the experiments - the one in Table 1. We simulated $2,500$ pairs playing 40 rounds against each other. We varied the maximal length of the considered plans, $K \in \{1, 2, 3, 4\}$, to see how that affected the behavior of the agents.

We classified the joint strategies of each pair in exactly the way we did for the data from the experiments. The results using this classification are contained in Table 3.

While IEL agents play the 40 round repeated Battle of the Sexes game in many different ways, we provide a couple of examples as illustrations of the way they play when $K = 2$.[12] In Figure 2 they converge to a Nash Equilibrium. In Figure 3 they converge to Alternate.

The results differ significantly as the maximal length of the considered plans changes.

K	% of Pure Nash	% of Alternation	% of Other
1	96	0	4
2	70	5	25
3	52	1	47
4	37	1	62

Table 3: Results of IEL behavior for various maximal plan lengths K

For $K = 1$, IEL agents almost always zero in on a Nash equilibrium. This is not surprising since they cannot consider more complex plans like alternation. This is also consistent with all the simulations from our past work, where $K = 1$ and IEL agents almost always converge rapidly to a Nash equilibrium.

For $K = 2$, 5% of IEL agents learn how to alternate. Considering length 2 plans allows some pairs to arrive at an alternating equilibrium. But, agents do not just switch from playing Nash to Alternate. 21% of those who played Nash when $K = 1$ now play Other when $K = 2$. Considering longer plans has caused some confusion and difficulty in coordinating.

For $K > 2$, it gets worse. As the maximal length of considered strategies increases, the percentage of Other increases. It becomes harder and harder for IEL agents to coordinate their strategies.

To summarize, an IEL agent is able to find simple plan equilibria, but has trouble coordinating on an alternating plan. The most coordination occurs if agents do not consider very complex plans and stick to plans of length 2. But, even then, IEL agents do not alternate as much as human subjects did in the experiments reported in Table 2.

[12] These examples were generated by beginning a simulation and taking the first instance of each. We could have generated a lot and then picked out the ones that "looked the best", but we decided that going with the first gives the reader a better idea of how IEL really performs.

Given the results of the experiments and simulations in this section, it would seem that we should reject IEL learning as an explanation of human behavior in Battle of Sexes games. However, the environment for the experiments we ran differs significantly from the simulations we ran. In particular, the information available to the IEL agents was different than that for the humans. The only information about the opponent that an IEL agent had in playing the BoS game was the actions that opponent took. An IEL agent did not know the information about the opponent's payoffs.

So before we reject IEL as an explanation of behavior in Battle of Sexes games, we need to look at experiments with information conditions similar to those of the IEL agents. We turn to that now.

3 Modifying the Experiment

To provide an environment for human subjects that was closer to that modeled with IEL, we ran a set of experiments in which subjects were only informed about their own payoffs. We refer to that as Dark information. This is similar to what Van Huyck et al. (2007a) did. They investigated whether behavior in a median effort coordination game changes when subjects are limited to the information used by reinforcement learning algorithms. They found that, in the experiment, subjects converge to an absorbing state at rates that are orders of magnitude faster than reinforcement learning algorithms, but slower than under complete information. As we will see below, IEL is much faster than reinforcement learning, and this is why we are able to compare the IEL to the experimental behavior.

To further confound those subjects who might want to assume that their opponents had the same payoffs as themselves, we switched to an asymmetric payoff matrix. See Table 4.

	1	0
1	3, 3	9, 17
0	20, 10	3, 3

Table 4: Asymmetric Payoff Table for BoS

This matrix is distinguished from the earlier symmetric payoff matrix in two ways. It is asymmetric, giving the row player a slightly higher payoff to their preferred Nash equilibrium. It strengthens the incentives towards the Nash equilibria and away from alternation. Now a player will gain 10 or 8 per period by playing their best Nash equilibrium over alternation as opposed to only 6 with the previous payoffs.

The rest of our experimental setup was exactly the same as in Section 2.1. [13]

[13] We provide all of our experimental data in the Supplemental Material.

3.1 Results

Using the same categorization as in Section 2.1, with asymmetric payoffs we find the results in Table 5.

plan	%	period started
Nash Equilibrium	65	10.4
Alternate	5	18
Other	30	na

Table 5: Experiment Results: Asymmetric, Dark

This treatment (Asymmetric payoffs in the Dark) eliminates much of the coordination other researchers find in Battle of Sexes games. To see whether the behavior of IEL is consistent with that of humans in this information condition, we ran the same simulations we did in Section 2.2.2 with the new payoff matrix given in Table 4. The results are presented in Table 6.

K	% of Pure Nash	% of Alternation	% of Other
1	91	0	9
2	64	6	30
3	44	1	55
4	28	0	72

Table 6: Results of IEL behavior for various K
and asymmetric payoffs

The behaviors of human subjects in the Dark - asymmetric payoffs treatment and IEL agents with $K = 2$ are virtually identical. For both humans and IEL, roughly 2/3 of the pairs play the stage game Nash Equilibrium and about 5% of the pairs Alternate.[14]

It was tempting at this point to declare victory. But that would ignore the experimental results under full information in Table 2. Is there a modification of IEL that would explain those? We turn to that now.

4 Modifying IEL

In Figure 4, we provide the pattern of play for two human subjects that is typical of the many times in the lab that players choose quickly to alternate. In this typical pattern, one player (here it is player 1) begins to alternate from the beginning. Clearly player 1 has not learned to alternate. Instead player 1 has begun play with a plan to alternate already in mind. The second player quickly sees what

[14] We admit that this was a particularly lucky outcome for us. We would not expect to get the exact same proportions if we replicated the experiment with another 20 pairs. But we would expect to get something similar.

is happening and joins in the alternation (with one exception in period 11). The second player has learned to alternate.

In this section we allow the IEL agents to use information about the game from the beginning. We let some of the players *have a good idea* after they see the payoff structure. That is, we allow the IEL agents to begin play with information more similar to the full information treatment in the experiments. We do this by populating the initial set of remembered plans with good ideas.[15] We place only copies of the alternating plan $(0, 1)$ in the initial set of remembered plans, $\mathcal{P}_1^i$. This is an agent who is convinced that the pair should follow alternating plans, moving from $(0, 1)$ to $(1, 0)$ to $(0, 1)$ and so forth. That is the only change to the IEL model of section 2.2.

Inspired agents start with their preferred Nash Equilibrium action, 0, and then move to 1. They expect the other player to learn to play this way also. If the other player does learn, the $(0, 1)$ plan will attain a high payoff and stay in the remembered set. But there is no guarantee that these "good ideas" will survive throughout the game. After the first period, the set of considered plans is subjected to the usual experimentation and replication. Therefore, a "good idea" will be replaced if it turned out to be a bad idea; that is, if it does not prove to be good enough against the actions the other player is using.[16]

For illustration, we show in Figures 5 and 6 two of the myriad ways an inspired IEL agent and an uninspired IEL agent might play against each other. In Figure 5, the uninspired agent learns and they converge to Alternate. In Figure 6, the uninspired agent does not learn and the agents converge to the preferred Nash Equilibrium of the uninspired agent. That is the inspired agent gives up faster than the uninspired agent learns. In our simulations reported below, this happened only once in 1,000 pairs.

4.1 Results

In Table 7 and Table 8, we present the results of simulations, playing each of the two types of inspiration against each other. We use $K = 2$, because that was the value that fit the experiment results in Table 5. All other parameter values of IEL remain the same as we used in the simulations in section 2.2. Each new simulation ran 2,500 times and each run consisted of 40 rounds.[17]

Using the same classification scheme as before, we counted the percentage of joint plans that were Nash, Alternate, or Other. We present the results for symmetric payoffs in Table 7 and for asymmetric payoffs in Table 8. In each cell, we present the rounded percentages of (Nash, Alternate, Other).

The types of joint strategies that IEL agents choose are clearly dependent on both the payoffs and the mix of inspiration.

[15] In this paper we do not model how these good ideas come into being. One possibility could be from a game theoretic analysis. Another could be social learning through prior experience such as modeled in Hanaki et. al. (2005).

[16] But this replacement will not happen instantaneously. Thus, this model is also consistent with the agent beginning with no inspiration and having an 'aha' moment sometime in the early rounds.

[17] When an uninspired agent plays another uninspired agent with symmetric payoffs, it is exactly the same as the simulation in Section 2.2 when $K = 2$. We use those results here.

Percentage of (Nash, Alternate, Other)

	uninspired	inspired
uninspired	70, 5, 25	0, 91, 9
inspired		15, 61, 24

Table 7: Simulation Results
with Symmetric Payoffs

Percentage of (Nash, Alternate, Other)

	uninspired	inspired
uninspired	65, 5, 30	0, 89, 11
inspired		11, 58, 31

Table 8: Simulation Results
with Asymmetric Payoffs

The inspired can and do teach the uninspired to alternate. This is not surprising. What is interesting is that an uninspired agent is more likely to learn from an inspired agent than is an inspired agent. Inspired IEL agents have problems with other inspired agents because all inspiration is in the form of playing (0,1). If both players start with and continue to play (0,1), it leads to a sequence of actions: (0,0), (1,1,), (0,0), (1,1) and so on. Each player wants to end up at their preferred Nash Equilibrium and this generates a low payoff to both players. If this continued forever it would be classified as Other. But that does not happen. Experimentation and Replication add plans that are better for the players. So experimentation plus replication can lead eventually to one player having mostly (0,1) in their remembered set and the other player having mostly (1,0) in their remembered set. This obviously occurs more rapidly the fewer initial (0,1) that one of the agents has. That is, the fully inspired do better with those that are not inspired than those who are.

The results with asymmetric payoffs are similar to those with symmetric payoffs but asymmetric payoffs clearly cause some problems for coordination. There is a decrease of about 10% in Nash equilibrium play and a reduction of 3% in alternation. Asymmetry increases conflict and leads to more Other type disagreements.

Of more importance, how does this new version of IEL with inspiration compare to experimental data?

5 Comparison of IEL simulations and lab results

As we will see below, the modification of IEL that allows for some initial inspiration is able to explain the laboratory data. But to make our task a bit harder, we first completed the natural 2x2 experimental design to include all combinations of payoff (symmetric and asymmetric) and information (full and dark).

5.1 Completing the Experimental Design

We completed the 2x2 experimental design by adding the treatments of Asymmetric-Full and Symmetric-Dark. The procedures were identical to those in Section 2.1.1 and Section 3 with one exception. We had 21 subjects in the Symmetric-Dark treatment.

The results from these new experiments, together with the previous results from Tables 2 and 5, are displayed in Table 9.

Full

		Symmetric		Asymmetric
plan	%	period started	%	period started
Nash Equilibrium	30	5.5	55	8.3
Alternate	40	8.25	40	8.4
Other	30	na	5	na

Dark

		Symmetric		Asymmetric
plan	%	period started	%	period started
Nash Equilibrium	52	10.8	65	10.4
Alternate	29	7.5	5	18
Other	19	na	30	na

Table 9: Experimental Results
Four Treatments

Both the information and payoff conditions have an effect on how humans play a repeated Battle of Sexes game. For both payoff conditions, Dark decreases Alternate and increases Nash Equilibrium play. Both the absolute and relative effects on Nash Equilibrium versus Alternate are larger with symmetric payoffs. Asymmetry increases Nash Equilibrium play and Dark, decreases Alternate.

For the Full information treatments, those pairs who settle on either Nash or Alternate do so fairly quickly although alternation takes a little longer. In the Dark, those picking Nash equilibrium seem to try other things first and then finally give in. In the Dark with symmetric payoffs, those who decide to alternate do so reasonably quickly. In the Dark with asymmetric payoffs, only one pair chose to alternate and it took them a long time to do that.

The results in the Dark were a bit surprising to us. In the Dark, subjects do not know whether payoffs are symmetric or asymmetric. A simple conjecture would be that there is no difference in the percentages of Nash and Alternate. However, that did not happen. There is an explanation but it is better discussed below after we compare the IEL simulations with the experimental results.

5.2 The Comparison

The methodological hurdle in comparing IEL behavior to human behavior is that we do not know which, if any, of the human subjects in the experiment began with inspiration. So, we think of the human data as being generated from random selections of pairs from a population with certain percentages of the two types. For each distribution of those types, a percentage of Nash Equilbria and

Alternate will arise. For example, suppose that 30% of the players are inspired. Then in 9% of the pairs both players will be inspired. In 49% of the pairs neither player will be inspired. In the other 42% of the pairs one will be inspired and one will be uninspired. Thus with that 30% of the players, for the symmetric payoff - see Table 7, we should see an average of $0.09(15) + 0.49(70) + 0.42(0) = 35.5\%$ Nash equilibrium pairs.

In Figure 7, we display the range of possible percentages of types of equilbria for both the symmetric and asymmetric payoffs. There are really two continuous curves but we have only displayed the numbers for percentages that are multiples of 5. The X's are those points for the symmetric payoffs and the diamonds are those points for the asymmetric payoffs. We also display, as single points, the experimental percentages of types of equilibria for both Full and Dark information.

Three observations can be made from the figure. First, IEL with inspiration is not a theory of everything. For both symmetric and asymmetric payoffs, as the percentage of inspired agents increases from 0 to 100%, the predicted percentages of Nash and Alternate move from the lowest southeast point on the curve to the northwest, eventually curving back around slightly south and then southeast. Inspiration essentially introduces only one new parameter. Second, IEL with inspiration explains the experimental percentages for BoS in the Dark very well. With both symmetric and asymmetric payoffs, the experimental data are right on the respective curves. Third, IEL with inspiration does less well in explaining the experimental data under Full information. But, the fact that the experimental data are not exactly on the relevant curve does not reject IEL as an explanation for subject behavior in Battle of Sexes experiments. To see why, we need a closer look at the numbers.

To determine how well IEL with inspiration explains the data, we first determined, separately for each of the four treatments, that percentage of inspiration that best explained the data. To do so, for each of the four treatments, we chose the percentages of inspiration to minimize the RMSD (root mean squared deviation) between the experimental data and the simulations. The RMSD is $\sqrt{\frac{\sum_{i=1}^{3}(x_i - y_i)^2)}{3}}$ where $i \in \{$ Nash, Alternate, Other $\}$, x_i are the percentages from the simulations and y_i are the percentages from the experiments. Percentages are expressed as 15 and not 0.15. For example, if p is the percentage of players inspired and we consider the Symmetric Dark treatment the RMSD $= \sqrt{(A^2 + B^2 + C^2)/3}$ where $A = [(70p^2 + 15(1-p)^2 - 52]$, $B = [(5p^2 + 91 * 2p(1-p) + 61(1-p)^2 - 29]$, and $C = [(25p^2 + 9 * 2p(1-p) + 24(1-p)^2 - 19]$. This is minimized at $p = 15$.

The sizes of the minimal RMSDs and the minimizing percentages are displayed in Table 10. The RMSDs correspond in relative size to our earlier observation, about Figure 7, that the experimental data are "on" the appropriate curve for the Dark treatments but not for the Full. In Table 11 we display the percentages of types of equilibria implied by the minimizing percent of inspiration and compare those to the percentages of types of equilibria found in our experiments.

treatment	uninspired	inspired	RMSD
Symmetric Dark	85	15	1.38
Asymmetric Dark	100	0	0.82
Symmetric Full	75	25	8.7
Asymmetric Full	82	18	13.82

Table 10: Size of RMSD for different treatments

Sym. Dark

	Nash	Alter.	Other
IEL	51	28	21
Experiment	52	29	19

Asy. Dark

	Nash	Alter.	Other
IEL	64	6	30
Experiment	65	5	30

Sym. Full

	Nash	Alternate	Other
IEL	40	41	19
Experiment	30	40	30

Asy. Full

	Nash	Alternate	Other
IEL	44	32	24
Experiment	55	40	5

Table 11: IEL with inspiration vs. Experimental behavior

It is difficult to know whether RMSDs of 9 or 14 are small or not. To get a better feeling for what these values of RMSD really mean, we ran a series of Monte Carlo simulations using IEL and then computed some statistics. For each of the treatments, we began with the minimizing percentage of inspiration in Table 10. We then drew 20 types of pairs at random using that minimizing percentage. Each pair played 40 rounds. At the end we computed the percentage that converged to Nash and to Alternate. This procedure gave us one observation equivalent to one of our experimental sessions. We did this 400 times to generate a distribution over the pairs of average percentages (Nash, Alternate) that we might see in one laboratory session. We report statistics from those distributions in Table 12. The Nash mean is the mean of the distribution of the percentage of Nash equilibrium in each observation.

Treatment	Sym-Dark	Asym-Dark	Sym-Full	Asym-Full
Nash mean	52.6	67.2	40.9	45.7
Nash variance	1.3	1.1	1.2	1.4
Alternate mean	28.8	4.3	40.8	29.5
Alternate variance	1.1	0.2	1.1	1.2
Covariance	-0.8	-0.1	-0.8	-0.8

Table 12: Monte Carlo statistics: All four treatments

We then computed the Mahalanobis distance[18] between the mean of the experimental observations of (% of Nash, % of Alternate) and the Monte Carlo distribution of of the IEL observations. Quoting from Wikipeia, the most intuitive of the explanations, "The Mahalanobis distance ... is a multi-dimensional generalization of how many standard deviations a point is away from the mean of a distribution. ... along each principal component axis, it measures the number of standard deviations from the point to the mean of the distribution."

[18] The Mahalanobis distance is $d(\overrightarrow{x}, \overrightarrow{y}) = \sqrt{((\overrightarrow{x} - \overrightarrow{y})^T S^{-1}(\overrightarrow{x} - \overrightarrow{y})}$, where $\overrightarrow{x}$ is the data point, $\overrightarrow{y}$ is the mean of the distribution, and S is the covariance matrix of the distribution. This is basically a variance adjusted RMSD. See Mahalanobis (1936) and McLachlan (1992) for definitions and uses of the Mahalanobis distance.

In Table 13, we provide the underlying data for the computation, the value of the Mahalanobis distance, M, and the probability that the distance of any IEL observation would be less than that of the experimental observation, Prob($x < M$). Assuming the Monte Carlo distribution is Normal (which it probably is not) we compute the probability that the distance of any observation would be less than this.[19] Under normality, the smaller that number, the more likely it is that our experimental observation could have come from the distribution.

Treatment	Sym-Dark	Asym-Dark	Sym-Full	Asym-Full
Experiment Nash	52	65	3	55
Nash mean	52.6	67.2	40.9	45.7
Experiment Alternate	29	5	40	40
Alternate mean	28.8	4.3	40.8	29.5
Mahalanobis distance	0.06	0.226	1.48	2.08
Prob($x < M$)	0.00	0.03	0.67	0.89

Table 13: Monte Carlo results vs Experimental data

For the Dark treatments, the very low RMSD and Prob($x < M$) confirm that IEL with inspiration explains that experimental behavior. One possible objection to that arises by recognizing that, at the very beginning of an experiment, all subjects see much the same thing whether it is a symmetric payoff matrix or an asymmetric matrix. Only the entries differ. So it is not entirely obvious where the 15%, who believe alternate is a good idea, come from. It has been conjectured by many experimentalists that, in the Dark, subjects initially assume others' payoffs are the same as theirs; i.e., that the payoff matrix is symmetric.[20] This could lead some to think, as they would under Full information, that alternation is a good strategy. Of course, that conjecture would be wrong in the asymmetric payoff treatment. If some subjects begin with some inspiration but give it up quickly after a few rounds, the effect would be similar to lower or no inspiration from the beginning. Perhaps asymmetry just leads to a quicker retreat.

For the Full information treatments, the RMSD and Prob($x < M$) are larger than one would like and suggest that, although IEL with inspiration goes a good ways towards explaining that data, something else is also happening. Two possibilities are: fairness concerns - subjects get disutility if their average payoffs per round are lower that the other's - and stubbornness - humans resist learning rapidly, hoping the other will give in first. Although we do not yet have a formal way of including these in the IEL model, we believe they can provide some intuition for the experimental results. Under full information with symmetric payoffs, it is not too hard to imagine that players might want to have equal payoffs on average. But if one is stubborn and tries to end up at their favorite Nash, joint play could easily lead to an outcome of Other. If that is true, we would see more Other and fewer Nash than IEL would predict. With asymmetric payoffs, fairness is less obvious and players may be less stubborn and accept the other's favorite Nash sooner than they would even in the Dark. If this is true, we would see less Other and more Nash than IEL would predict. In both cases, this is exactly what we see in the data.

[19] For a normal distribution of 2 dimensions, the square of the distance of an observation, d^2, is chi-square distributed. So the probability that $d < t$ is $1 - \exp(\frac{-t^2}{2})$.

[20] We thank Catherine Eckel for reminding us of this observation.

6 Summary and Thoughts

We study when and how subjects in a repeated Battle of Sexes game learn to coordinate. We conducted experiments with human subjects, simulations with IEL agents, and compared their behavior in 4 different environments. We considered two information treatments: full information, in which subjects know each others payoffs, and dark information, in which subjects know only their own payoffs. We considered two payoff matrices: symmetric and asymmetric.

In the experiments with symmetric payoffs and full information, 40% of the pairs of subjects alternate and 30% settle on one of the Nash equilibria. The other 30% are confused or contentious. With asymmetric payoffs and dark information, only 5% alternate with 65% settling for Nash. 30% are confused or contentious. Both the Dark and Asymmetric treatments increase the percentage of Nash and lower the percentage of Alternation. Both the absolute and relative effects of Dark information are larger with Asymmetric payoffs.

In prior work we showed that the simple IEL model (with one period plans) worked very well at simulating how human subjects behave in a variety of different laboratory experiments. In this paper we provide a generalized theory of Individual Evolutionary Learning that allows agents to consider multi-period plans. The simulation results almost perfectly match the experiment results for the Dark -Asymmetric environment. We also provide a modification of IEL that incorporates strategic inspiration. This is the equivalent of adding one parameter. If 15% of the agents are inspired, the simulation results almost perfectly match the experiment results for the Dark -Symmetric environment. That is, IEL with strategic inspiration is an excellent explanation of when and how subjects who play in the Dark in a repeated Battle of Sexes game learn how to coordinate on a Nash Equilibrium or Alternation.

With the appropriate choice of levels of inspiration, the simulation results explain much but not all of the experimental results for the Full information conditions. What seems to be missing from the theory when there is Full information is some consideration of fairness and stubbornness.

Our results suggest that IEL is a superior model in terms of modelling multi period strategies. Unlike other algorithms that require an extremely long stage of 'pre-experimental' training, IEL simulations match experiments one-for-one in terms of the required number of periods. This suggests that it is worthwhile to explore the behavior of the multi-period IEL in other repeated game frameworks.

However, we are well aware that our modified IEL theory, while providing what looks to be a fairly good *ex post* explanation of human behavior in repeated Battle of Sexes games, does not provide the *ex ante* predictive model we would like to have. What is needed for that is an endogenous model of where the inspiration comes from. There seem to be two possibilities. (1) Some form of social learning either in prior experiments or in their experiences outside of the lab. (2) Some form of strategic game theoretic analysis of the situation. In both cases there must be room for heterogeneity in the extent of inspiration.

Having a good idea or having the capability to be inspired is something subjects bring with them to the lab. It is a part of their type that the experimenter does not usually control, like risk attitudes, or beliefs. One could, as we did in Arifovic and Ledyard (2012), estimate the distribution of these types in the subject pool. If the predictions of IEL were reasonably robust to that distribution, as was true in Arifovic and Ledyard (2012), and if the distribution of types in the population were stable, then this would be a reasonable approach. But, as explained by Cason et. al. (2013), the distribution of the types of inspiration based on social learning seems to be sensitive to things like past experience in turn-taking.[21] Inspiration from strategic thinking may also depend on the number of courses with game theoretic content that a subject has taken. Inspiration therefore is not a fixed characteristic. It is an interesting open question how to deal with this theoretically and in the lab.

Finally, we want to acknowledge the debt we owe to John Van Huyck's research. His work on coordination and equilibrium selection in games with multiple equilibria answered questions and raised issues that led to the work we report on in this paper. In particular, he often tried to compare the predictions of various adaptive algorithms and the outcomes of experiments in repeated games. In Van Huyck et. al. (1994), they considered a myopic best response dynamic and an inertial selection dynamic. In Van Huyck et. al. (2007a) they considered a reinforcement algorithm. Often subjects learned much faster than the algorithms which led them to conclude that "a realistic model of adaptive behavior would have to allow for heterogeneity and random exploration." The IEL model has both these features and thus, as they conjectured, is able to learn as fast as humans do and to produce results that are close to those produced by the humans.

[21] Also see Bednar et. al. (2017) for similar findings.

References

1. Arifovic, J., McKelvey, R.D., and S. Pevnitskaya (2006) "An Initial Implementation of the Turing Tournaments to Learning in Two Person Games", *Games and Economic Behavior*, 57, 93-122.
2. Arifovic, J. and J. Ledyard (2004) "Scaling up Learning Models in Public Good Games", *Journal of Public Economic Theory*, 6(2) 203-238.
3. Arifovic, J. and J. Ledyard (2007) "Call Market Book Information and Efficiency", *Journal of Economic Dynamics and Control*, 31, 1971-2000.
4. Arifovic, J. and J. Ledyard (2012) "Individual Evolutionary Learning, Other-regarding Preferences, and the Voluntary Contributions Mechanism", Journal of Public Economics, 96, 808-823. http://dx.doi.org/10.1016/j.jpubeco.2012.05.013
5. Arifovic, J. and J. Ledyard (2011) "A Behavioral Model for Mechanism Design: Individual Evolutionary Learning", *Journal of Economic Behavior and Organization*, Volume 78, Issue 3, 374-395. http://dx.doi.org/10.1016/j.jebo.2011.01.021
6. Bednar, J., Y. Chen, T. X. Liu, and S. Page (2017) "Behavioral spillovers and cognitive load in multiple games: An experimental study", to appear in *Games and Economic Behavior*.
7. Boylan, R. and El Gamal M., (1993) "Fictitious Play: A Statistical Study of Multiple Economic Experiments" *Games and Economic Behavior* 5, 205-222.
8. Bush, R. R. and Mosteller, F. (1951) "A mathematical model for simple learning", *Psychological Review* 58: 313-323. doi:10.1037/h0054388.
9. Camerer, C.F. and Ho, T. (1999) "Experience-Weighted Attraction in Games", *Econometrica*, 67: 827-874.
10. Camerer, C. F., T. Ho, and J. Chong (2002) "Sophisticated EWA learning and strategic teaching in repeated games." *Journal of Economic Theory*, 104, 137188.
11. Cason, T., S. Lau, and V. Mui, (2013) "Learning, teaching, and turn taking in the repeated assignment game", *Economic Theory*, 54, 335-357.
12. Cheung, Y., and D. Friedman (1997), "Individual learning in normal form games: some laboratory results." *Games and Economic Behavior*, 55, 340-371.
13. Erev, I. and A.E. Roth, (1998) "Predicting how people play games: reinforcement learning in experimetnal games with unique, mixed strategy equilibria", *American Economic Review* 88: 848-881.
14. Erev, I., Ert, E., and Roth, A. E. (2010) "A choice prediction competition for market entry games: an introduction. *Games* 1, 117136.
15. Fischbacher, U. (2007) "z-Tree: Zurich toolbox for ready-made economic experiments," *Experimental Economics*, 10, 171-178.
16. Fudenberg, D. and D. Levine (1998) *Theory of Learning in Games*, MIT Press, Cambdrige, MA.
17. Hanaki, N., R. Sethi, I. Erev, and A. Peterhansl (2005) "Learning plans", *Journal of Economic Behavior and Organization*, 56: 523 - 542.
18. Ioannou, C. and J. Romero (2014) "A generalized approach to belief learning in repeated games" , *Games and Economic Behavior*, 87, 178-203.
19. Mahalanobis, P. C. (1936) "On the generalised distance in statistics", *Proceedings of the National Institute of Sciences of India*, 2 (1): 49-55.
20. McLachlan, G. J. (1992) *Discriminant Analysis and Statistical Pattern Recognition*, Wiley Interscience, p. 12. ISBN 0-471-69115-1
21. McKelvey, R. D. and Palfrey, T.R. (2002) Playing in the Dark: Information, Learning, and Coordination in Repeated Games. Caltech Working Paper.
22. Myung, N. and J. Romero (2013) "Computational Testbeds for Coordination Games", Working Paper.
23. Rapoport, A., J.G. Melvin, and D.G. Gordon (1978) *The 2x2 Game*, University of Michigan Press.
24. Selten, R. and R. Stoecker (1986) "End behaviour in sequences of Finite Prisoner's Dilemma Supergames: a Learning Theory Approach", *Journal of Economic Behavior and Organization* 7, 47-70.
25. Sonsino, D. and J. Sirota, "Strategic pattern recognition - experimental evidence", *Games and Economic Behavior*, 44, 390-411, 2003.
26. Van Huyck, J.B., J.B, R.C. Battalio, and R.O. Beil (1990), "Tacit Coordination Games, Strategic Uncertainty, and Coordination Failure", *The American Economic Review*, 80, 234-248.
27. Van Huyck, J.B., R.C. Battalio, and R. O. Beil (1991) "Strategic Uncertainty, Equilibrium Selection, and Coordination Failure in Average Opinion Games" *The Quarterly Journal of Economics*, 106, 885-910.
28. Van Huyck, J.B., J.P. Cook and R.C. Battalio (1994) " Selection Dynamics, Asymptotic Stability, and Adaptive Behavior", *Journal of Political Economy* 102, 975-1005.
29. Van Huyck, J.B. J.P. Cook, and R.C. Battalio (1997) " Adaptive Behavior and Coordination Failure", *Journal of Economic Behavior and Organization*, 32, 483-503.
30. Van Huyck, J.B., R.C. Battalio, and F.W. Ranking (2007a) "Selection Dynamics and Adaptive Behavior without Much Information", *Economic Theory*, 33, 53-65.
31. Van Huyck, J.B. Battalio, and M.F. Walters (2007b) " Evidence on Learning in Coordination Games", *Experimental Economics* 10, 205 - 220.
32. Sargent, T. "Bounded Rationality in Macroeconomics". Oxford: Oxford University Press, 1993.

Figures

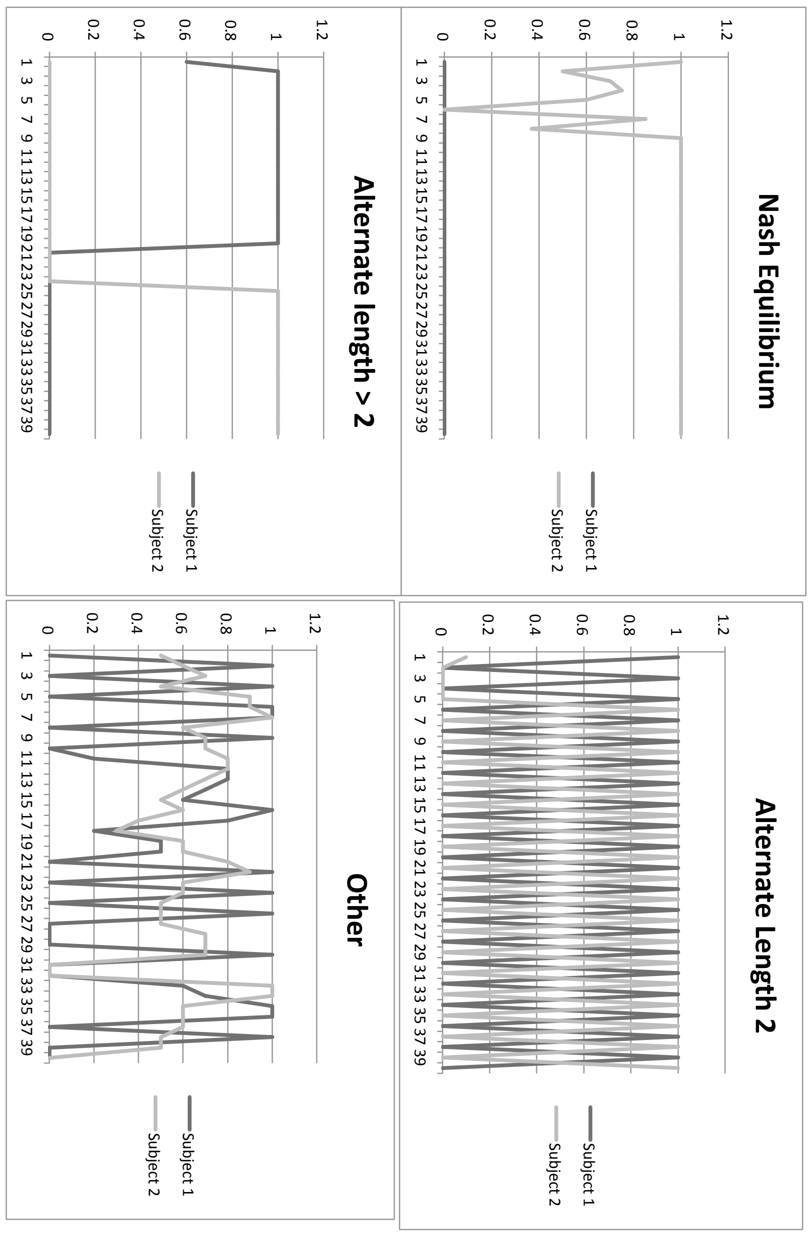

Fig. 1: Four joint stategies

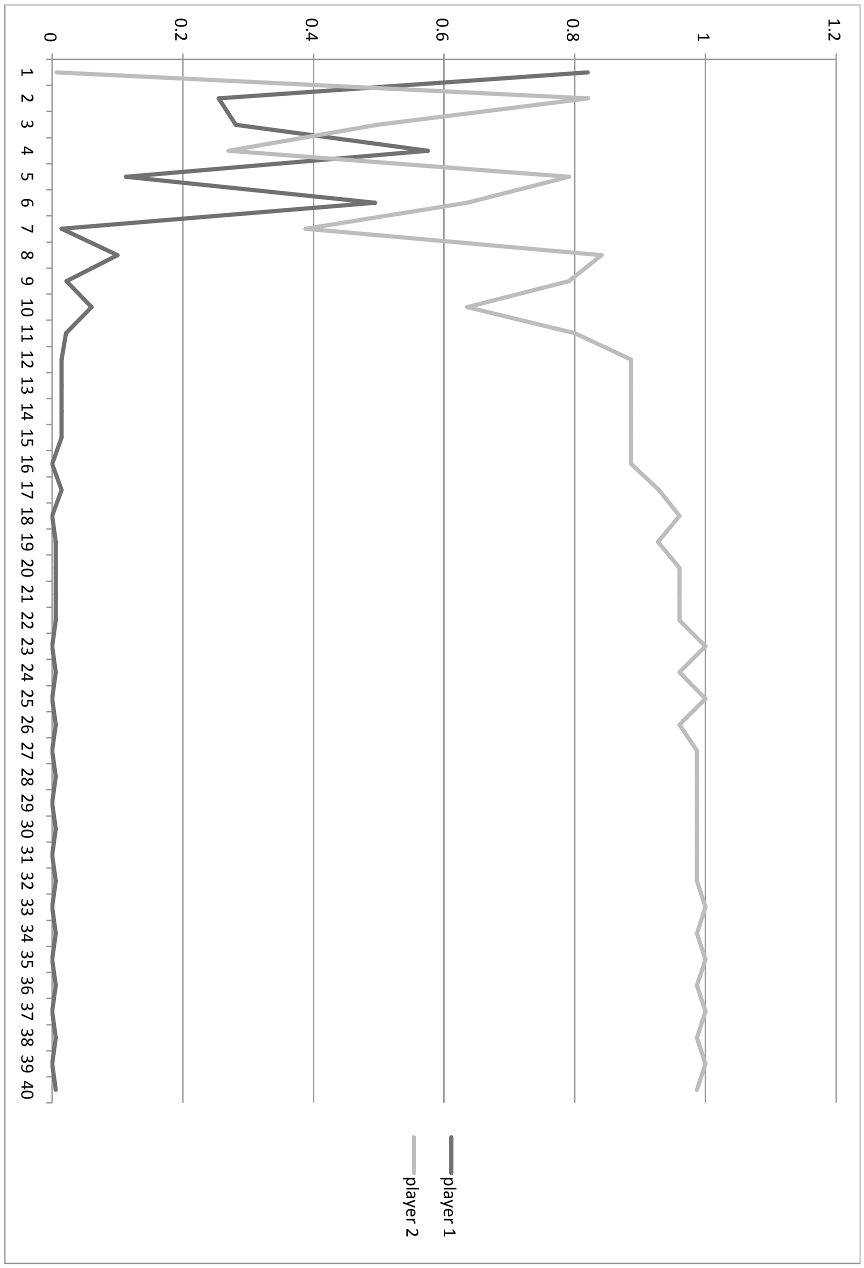

Fig. 2: Actions played by 2 IEL agents converge to Nash Equilibrium: $K = 2$.

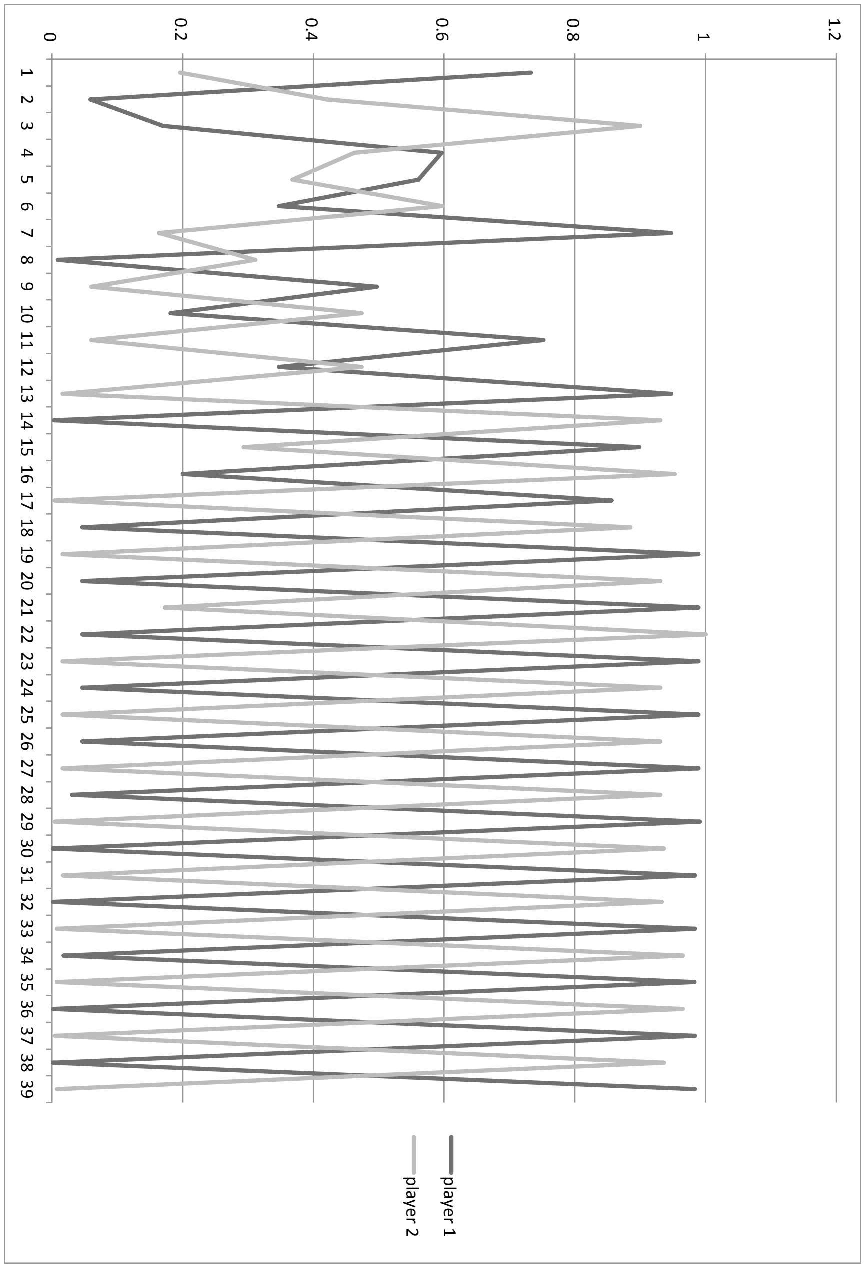

Fig. 3: Actions played by 2 IEL agents converge to Alternation: $K = 2$

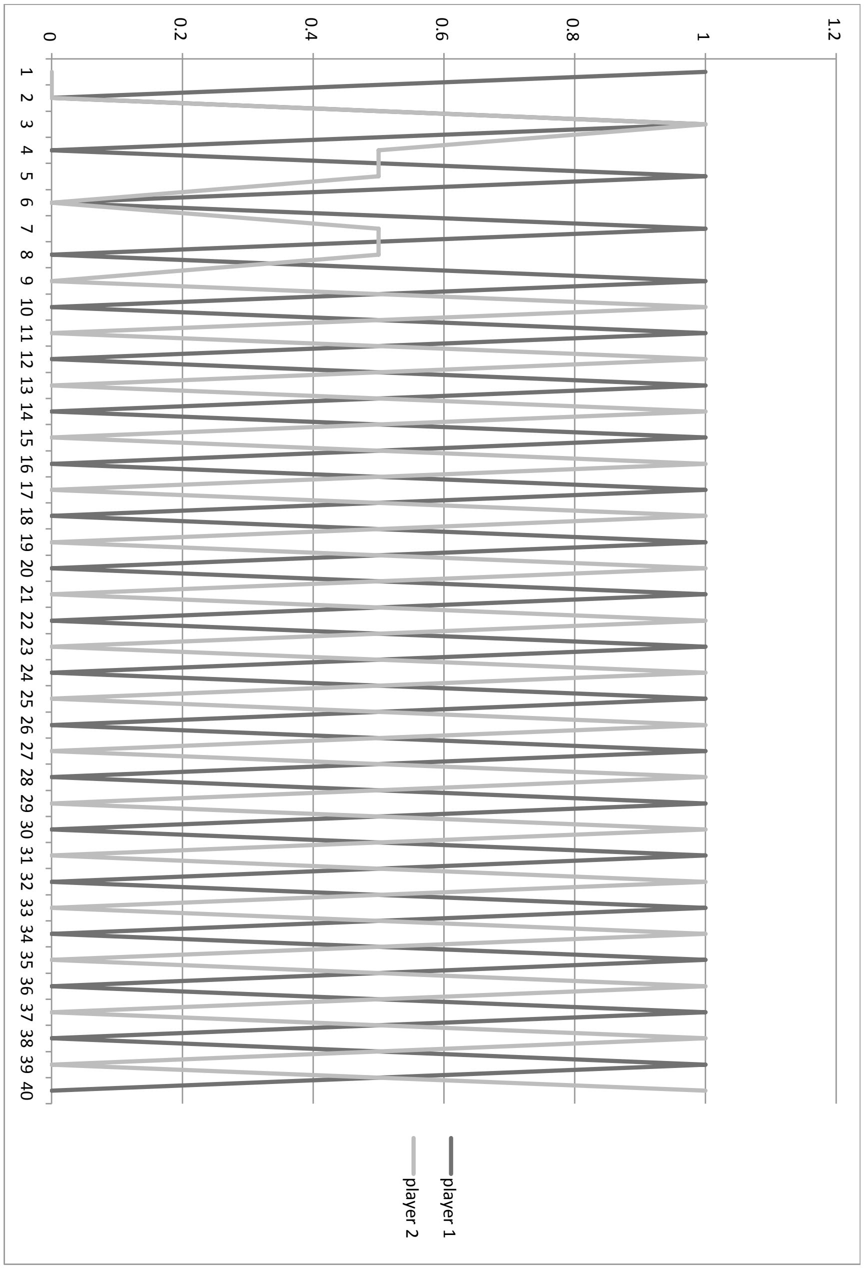

Fig. 4: One alternates - the other learns

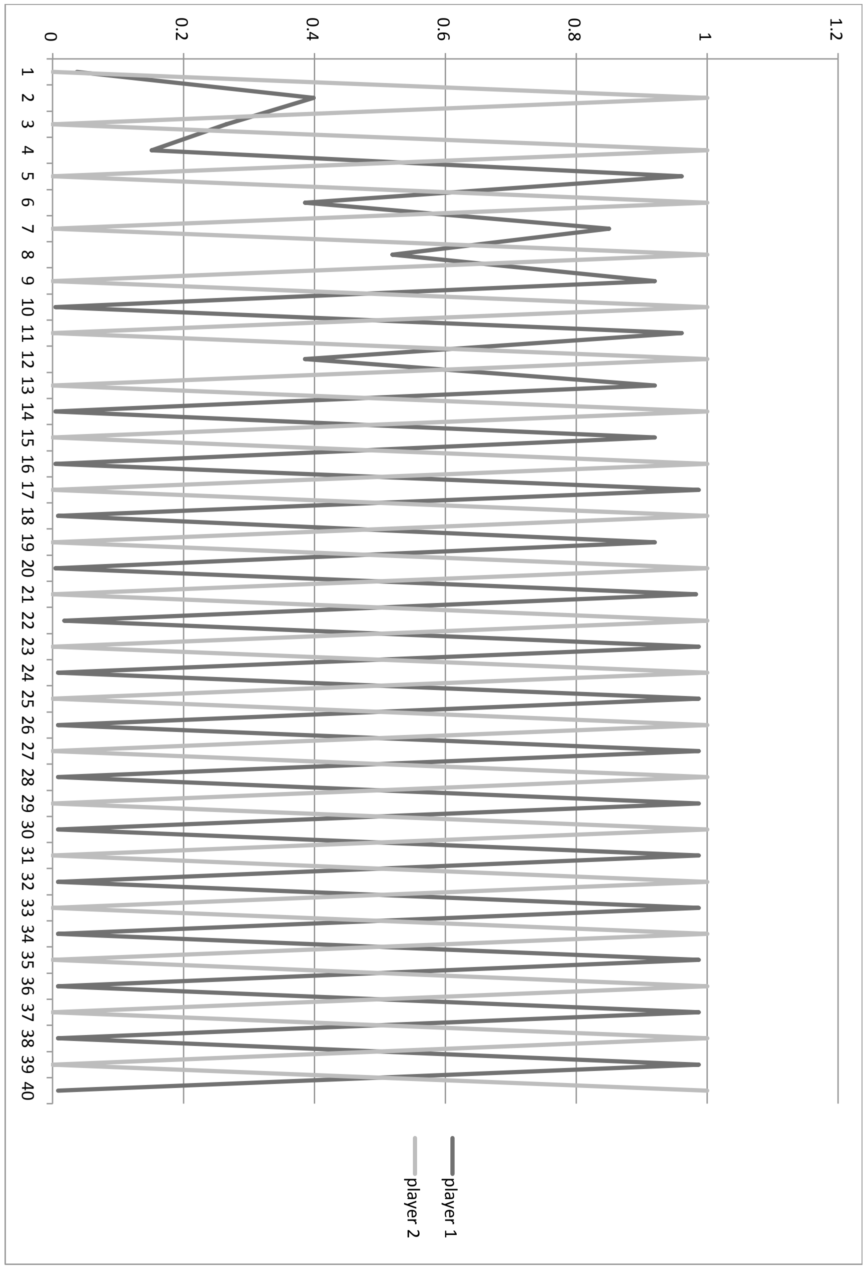

Fig. 5: Actions played by 2 IEL agents (Player 1 uninspired, Player 2 inspired) converge to Alternation: $K = 2$

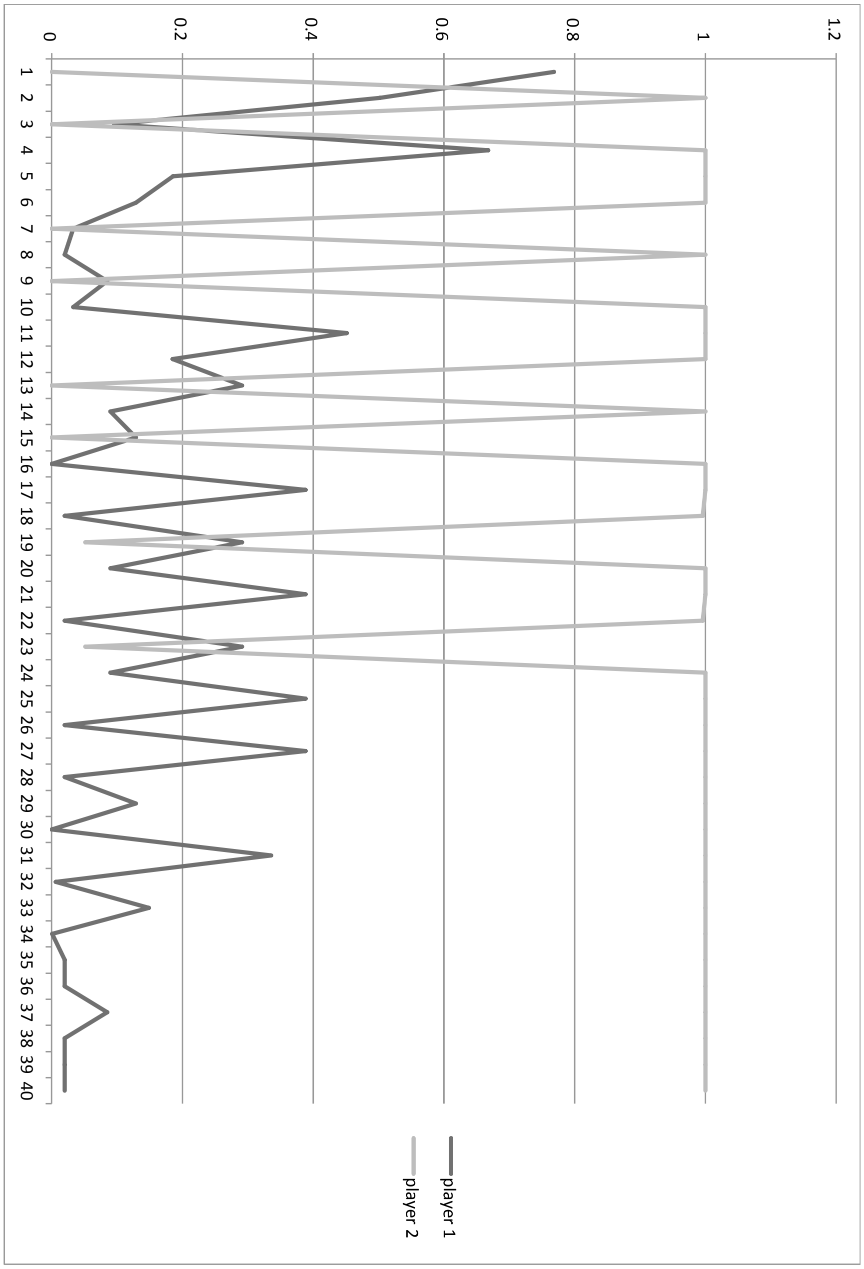

Fig. 6: Actions played by 2 IEL agents (Player 1 uninspired, Player 2 inspired) converge to Nash Equilibrium: $K = 2$

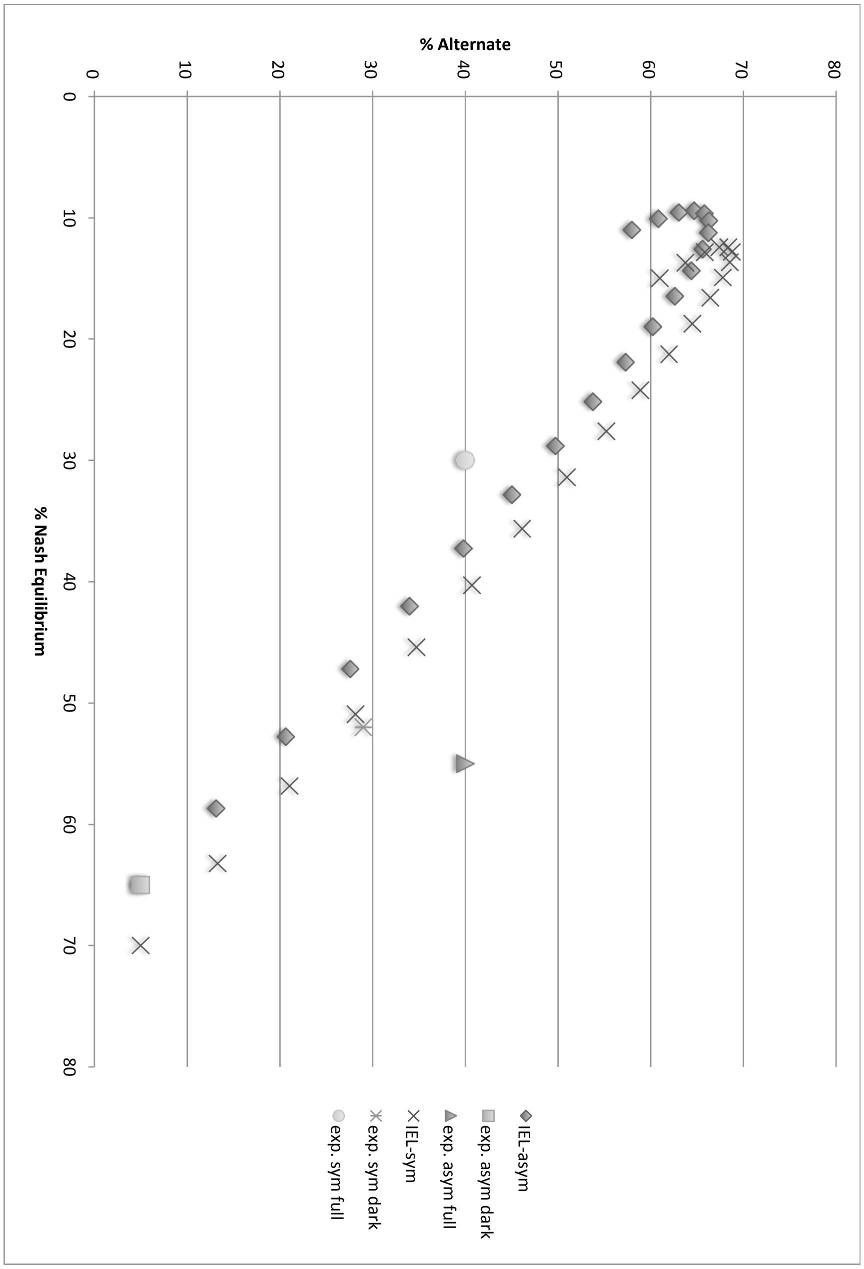

Fig. 7: IEL v. Experiment: All data

Appendix - Description of IEL

IEL is based on an evolutionary process which is individual, and not social. IEL is particularly well-suited to repeated games with large action spaces such as convex subsets of multi-dimensional Euclidean space. At the heart of each IEL agent's strategy is a collection of possible plans of action that they carry in their memory. These remembered plans are continually evaluated and the better ones are implemented with higher probability. In previous manifestations, IEL agents used particularly simple plans; those that involved only one period. In this paper, we allow them to consider multi-period plans.[22]

A *plan of action of length K for agent i* is $p^i = [p^i(1), ..., p^i(k)]$ where $p^i(r) \in A^i$. A finite *set of plans for i at t* is $\mathcal{P}^i_t = \{p^i_j\}_{j=1}^J$ where[23] $\mathcal{P}^i_t \subset A^{i(1)} \cup ... \cup A^{i(K)}$.

Initialization At the start of play, i's initial set $\mathcal{P}^i_0$ is created by randomly selecting J elements from $A^{i(1)} \cup ... \cup A^{i(K)}$. For $j = 1, ..., J$, a value for the length k is chosen from $\{1, ..., K\}$ with equal probability and, then, for each $p_j(d)$ with $d = 1, ..., k$, a value is randomly generated from $[0, 1]$. Agent i next selects a plan p_{0j} from $\mathcal{P}^i_0$ with probability $1/J$. Finally i implements the plan so selected. If the selected plan is j^* then $a^i_0 = p^i_{0,j^*}(1)$.

As the repeated game progresses, when a plan $p^i_{t,j} = (p^i_{t,j}(1), ..., p^i_{t,j}(k))$ is selected, it is implemented in its entirety by the IEL agent. That is, the entry $p^i_{tj}(1)$ is played in period t, the entry $p^i_{tj}(d)$ is played in period $t + d - 1$ up to period $t + k - 1$. An IEL agent's strategy implements a plan, lets it play out, and then chooses a new plan to implement. At the beginning of a round t in which i s not continuing implementation of a previously selected plan,i computes a new set of plans $\mathcal{P}^i_t$ and valuations W^i_t. This computation is at the heart of our behavioral model and consists of three pieces: *rotation*, *experimentation* and *replication*.

Rotation This step sets things up for the next round by repositioning every plan in the considered set. Every plan $p^i_{t-1,j} = (p^i_{t-1,j}(1), ..., p^i_{t-1,j}(k)) \in \mathcal{P}^i_{t-1}$, is replaced with a new plan $p^{*i}_{t-1,j} = (p^i_{t-1,j}(2), ..., p^i_{t-1,j}(k), p^i_{t-1,j}(1))$. That is all plans with length greater than one are rotated so as to remain consistent with the timing of play.

Experimentation Experimentation introduces new, alternative plans that otherwise might never have a chance to be tried. This insures that a certain amount of diversity in possible plans is maintained.

For each $j = 1, \ldots, J$, with probability ρ_{val}, $p^i_{t-1,j} \in \mathcal{P}^i_{t-1}$ is chosen for value experimentation. If j is not chosen for value experimentation, then $p^i_{tj} = p^i_{t-1,j}$. If j is chosen, then for all d, $p^i_{tj}(d) \sim N(p_{t-1,j}(d), \sigma | A^i)$. That is the new value for the dth entry of p is chosen randomly from the truncated normal centered at the old value. We now have a new, temporary set $X_t = \{p^i_{tj}\}_{j=1}^J$.

[22] As the reader will see, a plan is not a strategy. A stratgey is a function describing the choice of an action, or probability density over actions, contingent on history. A plan is simply a commitment to a sequence of actions, independently of what others will do. It may be part of the implementation of a strategy but it is not a strategy in and of itself. For example, "Tit for Tat" is a strategy while "Cooperate today and Defect tomorrow" is a plan.

[23] We use the symbol $X^{(k)}$ to denote the k times Cartesian product of X.

Next experimentation with length occurs. For each $j = 1, \ldots, J$, with probability ρ_{len}, $p^i_{j,t} \in X_t$ is expanded or contracted by 1 with equal probability if that is possible.[24] If j is expanded, the new value is appended at the latest time. The new element is either 0 or 1, whichever has the highest foregone utility.

Replication Replication follows experimentation. Replication reinforces plans that would have been good choices in previous rounds. It does that by creating a value for each plan and allowing higher valued plans to replace those that are valued less. Replication begins with the set X_t left after experimentation and creates a new set $\mathcal{P}^i_t$.

First, for each $p^i_{t,j} \in X_t$, compute valuations $W^i_{t,j}$. These valuations are based on the average utility that plan would have attained over the past L periods had it been played. For each p^i_{tj} in X_t

$$W^i_{t,j} = (1/L) \sum_{d=1}^{L} u^i(p^i_j(d), a^{-i}_{t-d+1-L}).$$

Here $a^{-i}_{t-d+1-L}$ is the action played by all other agents in period $t-d+1-L$. If $p^i_j = (p^i_j(1), ..., p^i_j(k))$ then the index z of $p^i_j(z)$ is mod k. In this paper we let $L = K$.

Next, for $j = 1, ..., J$, pick two members of $\{1, \ldots, J\}$ randomly (with uniform probability) with replacement. Let these be k and k'. Then,

$$p^i_{j,t} = \left\{ \begin{array}{l} p^i_{k,t-1} \\ p^i_{k',t-1} \end{array} \right\} \text{if} \left\{ \begin{array}{l} W^i_{t,k} \geq W^i_{t,k'} \\ W^i_{t,k} < W^i_{t,k'} \end{array} \right\}.$$

Note that $p^i_{j,t} \in \mathcal{P}^i_t$ while $p^i_{k,t-1}$ and $p^i_{k',t-1} \in X_t$. After $\mathcal{P}^i_t$ is created, the respective valuations W^i_j are computed for those elements.

Replication for t favors alternatives with a lot of replicates at $t-1$ and favors alternatives that would have done well over the past L periods, had they been used. So replication is a process with a form of averaging over *all* past periods. If the actual actions of others would have provided a favorable situation for particular plan p^i_j on average then that plan will tend to accumulate replicates in $\mathcal{P}^i_t$. It is fondly remembered and will be more likely to be actually implemented. Over time, the sets $\mathcal{P}^i_t$ will become more homogeneous as most of its plans become replicates of the best performing alternative.

After updating the set of plans and their valuations, the IEL agent selects a plan for t, p^i_t.

Selection The set of plans and their valuations, $(\mathcal{P}^i_t, W^i_t)$, induce a mixed plan on $\Delta(A^i)$ at t, and for some future rounds. In round t, i *selects* a plan $p^i_j \in \mathcal{P}^i_t$ with probability[25] $\pi^i_{t,j} = \frac{W^i_{t,j}}{\sum_{k=1}^{J} W^i_{t,k}}$. The selected plan is then implemented in its entirety.

[24] If $p^i_{t,j} = p^i_{t,j}(1)$ then j is only expanded. If $p^i_{t,j} = [p^i_{t,j}(1), ..., p^i_{t,j}(K)]$ then j is only contracted.

[25] If some of the $W^i_{t,j}$ are less than zero, we need to adjust this equation so as to produce a probability in $[0, 1]$. We let $\pi^i_{t,j} = \frac{W^i_{t,j} - \epsilon^i_t}{\sum_{k=1}^{J}(W^i_{t,k} - \epsilon^i_t)}$ where $\epsilon^i_t = \min\{0, \min_j W^i_{t,j}\}$.

Parameters IEL is completely specified by 5 parameters: $(J, \rho_{val}, \rho_{len}, \sigma, K)$. We let $J = 180$, $\rho_{val} = 0.033$, $\sigma = 0.05$, and $\rho_{len} = 0.0033$. Results from our previous IEL applications show that the same parameter values for $(J, \rho_{value}, \sigma)$ have worked well across many different environments (call market experiments in Arifovic and Ledyard, 2007; Groves-Ledyard experiments in Arifovic and Ledyard, 2004, 2011; voluntary contribution mechanisms, Arifovic and Ledyard, 2012). Further those results have also shown that the behavior of an IEL agent is reasonably robust against changes in these parameters.

We now have a complete model of behavior for a general repeated game.

Appendix - Instructions

Instructions for Dark Information

Introduction

You are about to participate in a session on decision making and you will be paid for your participation. This is part of a study intended to provide insight into certain features of decision processes. What you earn depends partly on your decisions and partly on the decisions of others. If you follow the instructions carefully and make good decisions you may earn a considerable amount of money. You will be paid in cash at the end of the experiment.

You will be asked to make choices in a number of rounds. You will be randomly paired with another person, your partner, for a sequence of rounds. A sequence of rounds is referred to as a **match**.

Match The length of a match is 40 rounds. In each round, you make a choice by entering a number between 0 and 1. The person you are matched with, your partner, also makes a choice. The payoff you receive for each round depends on your and your partners choices. For each round, you will see on the computer screen the information about your choice, your partners choice, and the payoff that you earned in that round.

Payoffs All the payoffs are stated in experimental currency units (ECU) and can be converted into cash at a rate of $1 dollar per 100 ECUs at the end of the experiment. The payoffs are accumulated over the rounds. In each round, you will see the cumulative payoff for all the rounds that you have already played at the bottom of the computer screen.

You can check your payoff for various combinations of your choices using the *payoff tables* that are in your folder or using the *what-if-calculator*.

Payoff Table For each match you can look up a payoff table to see what payoff you would receive for different choices that the two of you make. Your choices are given on the vertical axis and your partners on the horizontal axis. The choices are given in the increments of 0.1. Each cell gives your payoff for a given combination of choices.

For example, in case of match 1, if you choose 0.6 and your partner chooses 0.5, your payoff is 7.8.

What-if-Calculator You will find a what-if-calculator on your computer screen. You can use it at the beginning of each round to calculate your payoff for various combinations of your choices.

The payoff tables and the what-if-calculator provide the same type of information about your payoffs for various combinations of choices.

Instructions for Full Information

Introduction

You are about to participate in a session on decision making and you will be paid for your participation. This is part of a study intended to provide insight into certain features of decision processes. What you earn depends partly on your decisions and partly on the decisions of others. If you follow the instructions carefully and make good decisions you may earn a considerable amount of money. You will be paid in cash at the end of the experiment.

You will be asked to make choices in a number of rounds. You will be randomly paired with another person, your partner, for a sequence of rounds. A sequence of rounds is referred to as a **match**.

Match The length of a match is 40 rounds. In each round, you make a choice by entering a number between 0 and 1. The person you are matched with, your partner, also makes a choice. The payoff you receive for each round depends on your and your partners choices. For each round, you will see on the computer screen the information about your choice, your partners choice, and the payoff that you each earned in that round.

Payoffs All the payoffs are stated in experimental currency units (ECU) and can be converted into cash at a rate of $1 dollar per 100 ECUs at the end of the experiment. The payoffs are accumulated over the rounds. In each round, you will see the cumulative payoff for all the rounds that you have already played at the bottom of the computer screen.

You can check your payoff for various combinations of your choices using the *payoff tables* that are in your folder or using the *what-if-calculator*.

Payoff Table For each match you can look up a payoff table to see what payoff you would receive for different choices that the two of you make. Your choices are given on the vertical axis and your partners on the horizontal axis. The choices are given in the increments of 0.1. Each cell gives your payoff for a given combination of choices.

For example, in case of match 1, if you choose 0.6 and your partner chooses 0.5, your payoff is 7.8.

What-if-Calculator You will find a what-if-calculator on your computer screen. You can use it at the beginning of each round to calculate your payoff for various combinations of your choices.

The payoff tables and the what-if-calculator provide the same type of information about your payoffs for various combinations of choices.

This document contains some of the supplementary material for

Title: Learning to Alternate
Journal: *Experimental Economics*
Authors: Jasmina Arifovic and John Ledyard

The corresponding author is Jasmina Arifovic, Simon Fraser University, arifovic@sfu.ca

Included in the materials are a toy example of how the IEL simulations proceed, a copy of the python program used to generate an IEL simulation, and a copy of the python program used for the Monte Carlo simulations.

There is also an excel file with the experiment data.

A toy example of an IEL simulation.

The IEL parameters we use are: J = 3, K = 2, mu_v = 0.8, mu_l = 0.8,

The number of rounds = 3

This is round 0

The current strategy for player 0 is:
A = { [0.46 0.58] [0.79] [0.24] }
pi = [0.33 0.33 0.33]
The current plan = [0.24] and the current action = 0.24

The current strategy for player 1 is:
A = { [0.   0.36] [0.09 0.8] [0.09 0.06] }
pi = [0.33 0.33 0.33]
The current plan = [0.   0.36] and tfhe current action = 0.0

IEL for player 0 :
After update A= { [0.58 0.46] [0.79] [0.24] }
First is length experimentation.
Mutate the length of j= 1
The new plan is [0.79 1.]
Mutate the length of j= 2
The new plan is [0.24 1.]
After Length experimentation A= { [0.58 0.46] [0.79 1.] [0.24 1.] }
Next is value experimentation.
Mutate the value of j= 0
After value experimentation A= { [0.44 0.4] [0.79 1.] [
0.24 1] }
Next, update W = [3.64 8.99 8.99]
Next is replication.
For plan 0 j1= 2 j2= 2 : A 0 = [0.24 1.] , W 0 = 8.99
For plan 1 j1= 0 j2= 2 : A 1 = [0.24 1.] , W 1 = 8.99
For plan 2 j1= 0 j2= 0 : A 2 = [0.44 0.4] , W 2 = 3.64
After replication A= { [0.24 1.] [0.24 1.] [0.44 0.4] }
and W = [8.99 8.99 3.64]

IEL for player 1 :
After update A= { [0.36 0.] [0.8   0.09] [0.06 0.09] }
1 is continuing a plan. Further updating is unnecessary.

This is round 1

The current strategy for player 0 is:
A = { [0.24 1.] [0.24 1.] [0.44 0.4] }
pi = [0.42 0.42 0.17]
The current plan = [0.24 1.] and tfhe current action = 0.24

The current strategy for player 1 is:
A = { [0.36 0.] [0.8 0.09] [0.06 0.09] }
pi = [0.42 0.42 0.17]
The current plan = [0.36] and tfhe current action = 0.36

IEL for player 0 :
After update A= { [1. 0.24] [1. 0.24] [0.4 0.44] }
0 is continuing a plan. Further updating is unnecessary.

IEL for player 1 :
After update A= { [0. 0.36] [0.09 0.8] [0.09 0.06] }
First is length experimentation.
Mutate the length of j= 0
0 len=K The new plan is [0.36]
Mutate the length of j= 1
1 len=K The new plan is [0.8]
Mutate the length of j= 2
2 len=K The new plan is [0.06]
After Length experimentation A= { [0.36] [0.8] [0.06] }
Next is value experimentation.
Mutate the value of j= 0
Mutate the value of j= 1
Mutate the value of j= 2
After value experimentation A= { [0.36] [0.71] [0.12] }
Next, update W = [4.8 5.91 4.01]
Next is replication.
For plan 0 j1= 1 j2= 2 : A 0 = [0.71] , W 0 = 5.91
For plan 1 j1= 0 j2= 2 : A 1 = [0.36] , W 1 = 4.8
For plan 2 j1= 0 j2= 1 : A 2 = [0.71] , W 2 = 5.91
After replication A= { [0.71] [0.36] [0.71] }
and W = [5.91 4.8 5.91]

This is round 2

The current strategy for player 0 is:
A = { [1. 0.24] [1. 0.24] [0.4 0.44] }
pi = [0.42 0.42 0.17]
The current plan = [1] and tfhe current action = 1.0

The current strategy for player 1 is:

A = { [0.71] [0.36] [0.71] }
pi = [0.36 0.29 0.36]
The current plan = [0.36] and tfhe current action = 0.36

IEL for player 0 :
After update A= { [0.24 1.] [0.24 1.] [0.44 0.4] }
First is length experimentation.
Mutate the length of j= 0
0 len=K The new plan is [1]
Mutate the length of j= 1
1 len=K The new plan is [1]
After Length experimentation A= { [1] [1] [0.44 0.4] }
Next is value experimentation.
Mutate the value of j= 0
Mutate the value of j= 1
Mutate the value of j= 2
After value experimentation A= { [0.97] [0.99] [0.43 0.37] }
Next, update W = [5.72 5.72 5.56]
Next is replication.
For plan 0 j1= 0 j2= 1 : A 0 = [0.99] , W 0 = 5.72
For plan 1 j1= 2 j2= 0 : A 1 = [0.97] , W 1 = 5.72
For plan 2 j1= 0 j2= 0 : A 2 = [0.97] , W 2 = 5.72
After replication A= { [0.99] [0.97] [0.97] }
and W = [5.72 5.72 5.72]

IEL for player 1 :
After update A= { [0.71] [0.36] [0.71] }
First is length experimentation.
Mutate the length of j= 0
The new plan is [0.71 0.]
Mutate the length of j= 2
The new plan is [0.71 0.]
After Length experimentation A= { [0.71 0.] [0.36] [0.71 0.] }
Next is value experimentation.
Mutate the value of j= 0
Mutate the value of j= 1
After value experimentation A= { [0.74 0.] [0.41] [0.71 0.] }
Next, update W = [10.47 6.89 10.45]
Next is replication.
For plan 0 j1= 0 j2= 0 : A 0 = [0.74 0.] , W 0 = 10.47
For plan 1 j1= 0 j2= 1 : A 1 = [0.74 0.] , W 1 = 10.47
For plan 2 j1= 1 j2= 2 : A 2 = [0.71 0.] , W 2 = 10.45
After replication A= { [0.74 0.] [0.74 0.] [0.71 0.] }
and W = [10.47 10.47 10.45]

This game is over.

m1 + m2 = 1.28 m3 = 0.88

The python program for an IEL simulation

```
import random
from scipy.stats import truncnorm
import numpy as np
from sys import exit

# this section contains the basic IEL parameters and routines

J=180   #number of items in "considered set" S
muv = .033 # rate of mutation of value
sigmav=.05    # variance on the mutation of value # sigmav = float((su-sl))/10
mul = .0033       # rate of mutation of length
K =4    # max length of a "considered strategy"
L = K   # how far back to consider in computing foregone utility

#Types of inspiration:

type =[0,0] #type 1 is all "smart", type 2 is half "smart" half random.
                                #Any other produces  both random.
smart = [0,1] # What smart wants to play

#This initialization takes the types as given
def generalinitialization(I,J,K,sl,su,smart):
        W=[]
        St=[]

        for i in range(I):
                Sit=[]
                temp=[1]*len(range(J))
                W.append(temp)
                if type[i]==1:
                        Sit=[]
                        for j in range(J):
                                Sit.append(smart[:])
                elif type[i]==2:
                        if 2*(J/2)!=J:
                                exit("error J is not even")

                        else:
                                for j in range(J/2):
                                        Sittemp=[]
                                        k= random.randrange(K)
                                        for n in range(k+1):
```

```
                    Sittemp.append(random.uniform(sl,su))
                Sit.append(Sittemp)
                Sit.append(smart[:])

        else:
            for j in range(J):
                Sittemp=[]
                k= random.randrange(K)
                for n in range(k+1):
                    Sittemp.append(random.uniform(sl,su))
                Sit.append(Sittemp)

        St.append(Sit)

    return St,W

def choiceprobabilitiesfori(utilities):
    choicepiti=[]
    e=min(utilities)
    if e <= 0:
        for j in range(J):
            utilities[j] -= e-1
    sumw=sum(utilities)
    if sumw == 0:
        exit("error - sumw=0")
    for j in range(J):
        choicepiti.append(utilities[j]/float(sumw))
    return choicepiti


def selectionfori(some_list, probabilities):
    x = random.uniform(0, 1)
    cumulative_probability = 0.0
    for item, item_probability in zip(some_list, probabilities):
        cumulative_probability += item_probability
        if x < cumulative_probability: break
    return item
# item is the selected strategy

def foregoneutility(strategy,past_actions,player_name):
    #know L,t
    LL=min(L,t+1)
    v=0
    lenk = len(strategy)
    for d in range(LL):
        v+=utility(strategy[lenk-(d%lenk)-1],past_actions[t-d],player_name)
```

```
        return v/float(LL)

def Vexperimentationfori(strategyset):
                #value experimentation - know sigmav,sl,su,muv, mul
        for j in range(J):
                if onstrat[i]==0:
                        if random.uniform(0,1) < muv:
                                for k in range (len(strategyset[j])):
                                        centers = strategyset[j][k]
                                        r = (truncnorm.rvs((sl-centers)/float(sigmav),
                                                        (su-centers)/float(sigmav),
                                                        loc=centers, scale = sigmav, size
=1))
                                        strategyset[j][k] = np.array(r).tolist()[0]
        return strategyset

def Lexperimentationfori(strategyset,past_actions,player_name):
                #lengthexperimentation
        if K >1:
                for j in range(J):
                        if random.uniform(0,1) < mul:
                                if len(strategyset[j])==1:
                                        alt=strategyset[j][:]
                                        alt.append(1)
                                        strategyset[j].append(0)

        W0=foregoneutility(strategyset[j],past_actions,player_name)

        W1=foregoneutility(alt,past_actions,player_name)
                                        if W1>W0:
                                                strategyset[j]=alt
                                elif len(strategyset[j])==K:
                                        strategyset[j].pop(0)
                                        strategyset[j]
                                else:
                                        if random.uniform(0,1) < .5:
                                                strategyset[j].pop(0)
                                        else:
                                                alt=strategyset[j][:]
                                                alt.append(1)
                                                strategyset[j].append(0)

        W0=foregoneutility(strategyset[j],past_actions,player_name)

        W1=foregoneutility(alt,past_actions,player_name)
                                                if W1>W0:
```

```
                                        strategyset[j]=alt

    return strategyset


def replicatefori(strategyset,utilities):
            newS = [0]*J
            newW = [0]*J
            for j in range(J):
                j1=random.randrange(J)
                j2=random.randrange(J)
                newS[j]=strategyset[j2][:]
                newW[j]=utilities[j2]
                if utilities[j1]> utilities[j2]:
                    newS[j]=strategyset[j1][:]
                    newW[j]=utilities[j1]
            return newS,newW

            #(newS,newW) = (replicated strategies, corresponding
            #foregone utilities)

def updateWfori(Set,past_actions,player_name):
    W=[]
    for j in range(J):
        W.append(foregoneutility(Set[j],past_actions,player_name))
    return W

def updateStfori(Set):
    for strats in Set:
        item = strats.pop(0)
        strats.append(item)
    return Set
    #moves all strats in Set up by 1 (last played goes to the back)

#These are the parameters etc. for BoS

I=2  #number of subjects in a game

su = 1 #upper bound on the strategy set [sl,su]
sl = 0  #lower bound on strategy set [sl,su]

    #For BoS there is a basic 2x2 game matrix  [ 0payoffij, 1payoffij ] with 2
players

#pm = [ [0, 9,15,0],[0,15,9,0]]  #the symmetric payoff matrix
```

```
pm=[[3,9,20,3],[3,17,10,3]]   #the asymmetric payoff matrix

# a check for BoS
if len(pm) != I:
	exit("need utility parameters for all I")

alpha= ([[pm[0][3],pm[0][1]-pm[0][0],pm[0][2]-pm[0][0],pm[0][0]-pm[0][1]-
    pm[0][2]+pm[0][3]],[pm[1][3],pm[1][1]-pm[1][0],pm[1][2]-pm[1][0],
    pm[1][0]-pm[1][1]- pm[1][2]+pm[1][3]]]) # utility parameters


def utility(contemplated_action,actionvector, player_name):
		pactionvector=list(actionvector)
		pactionvector[player_name] = contemplated_action
		computed_utility =
(alpha[player_name][0]+alpha[player_name][1]*pactionvector[0]+

	alpha[player_name][2]*pactionvector[1] +

	alpha[player_name][3]*pactionvector[0]*pactionvector[1] )
		return computed_utility

# this is where the simulations start

# this section is for the simulation parameters

rounds=40   # number of rounds per run
runs = 1000  # number of runs in a  simulation

nash = 0 #initialize counts
alt = 0  #initialize counts

for sims in range(runs):

	random.seed()
	S=[]  #stores strategy sets for each run
	a=[]   #stores all actions for a run

#initialize for a run (i.e. one play of repeated game)
	onstrat=[0]*I
	currentstrat=[0]*I

	[St,W]= generalinitialization(I,J,K,sl,su,smart)

	for t in range(rounds):
		at=[]  #initializes actions for round t
```

```
        #this is the game play

                for i in range(I):
                        if onstrat[i] == 0:
                                p=choiceprobabilitiesfori(W[i])
                                currentstrat[i]=selectionfori(St[i],p)[:]
                        at.append(currentstrat[i][0])

                S.append(St)
                a.append(at)

        #this updates a player's "considered set"

                for i in range(I):
                        St[i]=updateStfori(St[i])
                        if len(currentstrat[i])>1:
                                del currentstrat[i][0]
                                onstrat[i] = 1
                        else:
                                onstrat[i]=0
                                St[i]=Lexperimentationfori(St[i],a,i)
                                St[i]=Vexperimentationfori(St[i])
                                W[i]=updateWfori(St[i],a,i)
                                (St[i],W[i])=replicatefori(St[i],W[i])

                #at this point we have S(t+1), W(t+1), onstrat(t+1), x(t+1).
                # we go to next t

        #record keeping before next simulation run
        temp = rounds -1
        m1 =(a[temp][0]-a[temp-1][0])**2 +(a[temp-2][0]-a[temp-3][0])**2
        m2 =(a[temp][1]-a[temp-1][1])**2 +(a[temp-2][1]-a[temp-3][1])**2
        m3=0
        for d in range(4):
                m3 += (a[temp - d][1]-a[temp-d][0])**2
        if (m1+m2 <1) and (m3> 3.5):
                nash +=1
                print a[temp][0],a[temp][1]
        if (m1+m2 >3) and (m3 >3.5):
                alt +=1

#reporting summary results of sims
print "nash = ",  nash/float(sims+1),
print "alternate =", alt/float(sims+1)
```

The python program for the Monte Carlo simulations

```
# Given a probability distribution on types, a number of pairs are matched
#and each plays a game.  That constitutes one observation (a block). Multiple
#blocks are run to generate a distribution.

import random
from scipy.stats import truncnorm
import numpy as np
from sys import exit

#IEL parameters

J=180   #number of items in "considered set" S
muv = .033 # rate of mutation of value
sigmav=.05    # variance on the mutation of value # sigmav = float((su-sl))/10
mul = .0033       # rate of mutation of length
K =2    # max length of a "considered strategy"
L = K   # how far back to consider in computing foregone utility

# IEL pieces

def generalinitialization(I,J,K,sl,su,smart):
	W=[]
	St=[]

	for i in range(I):
		Sit=[]
		temp=[1]*len(range(J))
		W.append(temp)
		if type[i]==1:
			Sit=[]
			for j in range(J):
				Sit.append(smart[:])
		elif type[i]==2:
			if 2*(J/2)!=J:
				exit("error J is not even")

			else:
				for j in range(J/2):
					Sittemp=[]
					k= random.randrange(K)
					for n in range(k+1):
						Sittemp.append(random.uniform(sl,su))
					Sit.append(Sittemp)
```

```
				Sit.append(smart[:])

		else:
			for j in range(J):
				Sittemp=[]
				k= random.randrange(K)
				for n in range(k+1):
					Sittemp.append(random.uniform(sl,su))
				Sit.append(Sittemp)

		St.append(Sit)

	return St,W

def choiceprobabilitiesfori(utilities):
	choicepiti=[]
	e=min(utilities)
	if e <= 0:
		for j in range(J):
			utilities[j] -= e-1
	sumw=sum(utilities)
	if sumw == 0:
		exit("error - sumw=0")
	for j in range(J):
		choicepiti.append(utilities[j]/float(sumw))
	return choicepiti

def selectionfori(some_list, probabilities):
	x = random.uniform(0, 1)
	cumulative_probability = 0.0
	for item, item_probability in zip(some_list, probabilities):
		cumulative_probability += item_probability
		if x < cumulative_probability: break
	return item

def foregoneutility(strategy,past_actions,player_name):
	#know L,t
	LL=min(L,t+1)
	v=0
	lenk = len(strategy)
	for d in range(LL):
		v+=utility(strategy[lenk-(d%lenk)-1],past_actions[t-d],player_name)
	return v/float(LL)

def Vexperimentationfori(strategyset):
		#value experimentation - know sigmav,sl,su,muv, mul
```

```
	for j in range(J):
		if onstrat[i]==0:
			if random.uniform(0,1) < muv:
				for k in range (len(strategyset[j])):
					centers = strategyset[j][k]
					r = (truncnorm.rvs((sl-centers)/float(sigmav),
							(su-centers)/float(sigmav),
							loc=centers, scale = sigmav, size
							=1))
					strategyset[j][k] = np.array(r).tolist()[0]
	return strategyset

def Lexperimentationfori(strategyset,past_actions,player_name):
		#lengthexperimentation
	if K >1:
		for j in range(J):
			if random.uniform(0,1) < mul:
				if len(strategyset[j])==1:
					alt=strategyset[j][:]
					alt.append(1)
					strategyset[j].append(0)

	W0=foregoneutility(strategyset[j],past_actions,player_name)
	W1=foregoneutility(alt,past_actions,player_name)
				if W1>W0:
					strategyset[j]=alt
				elif len(strategyset[j])==K:
					strategyset[j].pop(0)
					strategyset[j]
				else:
					if random.uniform(0,1) < .5:
						strategyset[j].pop(0)
					else:
						alt=strategyset[j][:]
						alt.append(1)
						strategyset[j].append(0)

	W0=foregoneutility(strategyset[j],past_actions,player_name)
	W1=foregoneutility(alt,past_actions,player_name)
						if W1>W0:
							strategyset[j]=alt

	return strategyset

def replicatefori(strategyset,utilities):
			newS = [0]*J
```

```
                    newW = [0]*J
                    for j in range(J):
                            j1=random.randrange(J)
                            j2=random.randrange(J)
                            newS[j]=strategyset[j2][:]
                            newW[j]=utilities[j2]
                            if utilities[j1]> utilities[j2]:
                                    newS[j]=strategyset[j1][:]
                                    newW[j]=utilities[j1]
                    return newS,newW

def updateWfori(Set,past_actions,player_name):
        W=[]
        for j in range(J):
                W.append(foregoneutility(Set[j],past_actions,player_name))
        return W

def updateStfori(Set):
        for strats in Set:
                item = strats.pop(0)
                strats.append(item)
        return Set

# this section is the basic game parameters used in these simulations (for BoS)
        # first the strategy set then the utility parameters and functions

I=2  #number of subjects in a game

su = 1 #upper bound on the strategy set [sl,su]
sl = 0  #lower bound on strategy set [sl,su]

#For BoS there is a basic 2x2 game matrix  [ 0payoffij, 1payoffij ] with 2 players

pm = [ [0, 9,15,0],[0,15,9,3]]  # pm is the payoff matrix

alpha= ([[pm[0][3],pm[0][1]-pm[0][0],pm[0][2]-pm[0][0],pm[0][0]-pm[0][1]-
    pm[0][2]+pm[0][3]],[pm[1][3],pm[1][1]-pm[1][0],pm[1][2]-pm[1][0],
    pm[1][0]-pm[1][1]- pm[1][2]+pm[1][3]]]) # utility parameters

def utility(contemplated_action,actionvector, player_name):
                pactionvector=list(actionvector)
                pactionvector[player_name] = contemplated_action
                computed_utility =
(alpha[player_name][0]+alpha[player_name][1]*pactionvector[0]+

        alpha[player_name][2]*pactionvector[1] +
```

```
        alpha[player_name][3]*pactionvector[0]*pactionvector[1] )
                return computed_utility

# this section is the basic IEL parameters used in these BoS simulations

#type =[0,0] #type 1 is all "smart", type 2 is half "smart" half random.
                        # type 0 is random.    Any other produces both random

smart = [0,1]

#probability any agent is type (0,1,2)
typeprobability= [.85,.15,0]

# this is where the simulations start

rounds=40   # number of rounds per run
runs = 200  # number of pairs in a  simulation
numblocks = 1

print "a block =", runs, "simulations"
print "there are", numblocks, "blocks"
print "probability type of (random, smart, half) =",
np.round(typeprobability,decimals=2)
print

print "block #, %Nash, %Alternate"
for block in range(numblocks):
        nash = 0 #initialize counts
        alt = 0  #initialize counts

        for sims in range(runs):

                random.seed()
                S=[]  #stores strategy sets for each run
                a=[]   #stores all actions for a run

                #Initialize for the type of each player in this pair
                z1=selectionfori([0,1,2], typeprobability)
                z2=selectionfori([0,1,2], typeprobability)
                type = [z1,z2]

#initialize for a run (i.e. one play of repeated game)
                onstrat=[0]*I
                currentstrat=[0]*I
                [St,W]= generalinitialization(I,J,K,sl,su,smart)
```

```
        for t in range(rounds):
        #print "round =", t
            at=[]  #initializes actions for round t

        #this is the round play

            for i in range(I):
                if onstrat[i] == 0:
                    p=choiceprobabilitiesfori(W[i])
                    currentstrat[i]=selectionfori(St[i],p)[:]
                at.append(currentstrat[i][0])

            S.append(St)
            a.append(at)

            for i in range(I):
                St[i]=updateStfori(St[i])

                if len(currentstrat[i])>1:
                    del currentstrat[i][0]
                    onstrat[i] = 1

                else:
                    onstrat[i]=0
                    St[i]=Lexperimentationfori(St[i],a,i)
                    St[i]=Vexperimentationfori(St[i])
                    W[i]=updateWfori(St[i],a,i)
                    (St[i],W[i])=replicatefori(St[i],W[i])

        #at this point we have S(t+1), W(t+1), onstrat(t+1), x(t+1).
        # we go to next t

#record keeping before next simulation run
        temp = rounds -1
        m1 =(a[temp][0]-a[temp-1][0])**2 +(a[temp-2][0]-a[temp-3][0])**2
        m2 =(a[temp][1]-a[temp-1][1])**2 +(a[temp-2][1]-a[temp-3][1])**2
        m3=0
        for d in range(4):
            m3 += (a[temp - d][1]-a[temp-d][0])**2
        if (m1+m2 <1) and (m3> 3.5):
            nash +=1
        if (m1+m2 >3) and (m3 >3.5):
            alt +=1
    print block, nash/float(sims+1),alt/float(sims+1)
```