

Computational Testbeds for Coordination Games*

Noah Myung[†]

Julian Romero[‡]

March 15, 2010

Abstract:

We propose a decision making process meant to mimic human behavior. This process is implemented with computational agents. We use this computational testbed to run simulations of two coordination games, the minimum-effort coordination game and the battle of the sexes game. We find that the computational agents exhibit behavior similar to human subjects from previous experimental work. We then use the computational testbed to develop experimental hypotheses, which are then confirmed in the laboratory using human subjects. In particular, we show that higher cost may actually lead to higher average payoffs in the minimum-effort coordination game.

JEL classification: *C63, C73, C91, C92*

Keywords: Battle of the Sexes. Coordination Game. Computational Testbed. Experiments. Learning. Minimum-effort coordination game.

* Myung's research was graciously funded by Jaksa Cvitanic while Myung was a student at Caltech and Romero's research was graciously funded by SISL. Many thanks John Ledyard, Leeat Yariv, and Jacob Goeree as well as seminar participants at Caltech, Graduate Workshop in Computational Social Science Modeling and Complexity, and ESA North America Meeting. Existing errors are our sole responsibility.

[†] Graduate School of Business & Public Policy, Naval Postgraduate School, Email: noah.myung@nps.edu. Web: <http://faculty.nps.edu/nmyung>

[‡] Division of Humanities and Social Sciences, California Institute of Technology, Email: jnr@hss.caltech.edu. Web: <http://jnromero.com>

1 Introduction

Economists have chosen two avenues to understand human interactions in repeated games. The theorists have focused on presenting learning algorithms and proving results on the convergence properties of these algorithms. Experimentalists have used human subjects in laboratory environments to better understand behavior in these interactions. In this paper, we develop a third avenue, computation, which complements both the theoretical and experimental literature. In particular, we build computational agents which exhibit behavior consistent with the behavior of human subjects in two repeated coordination games. Using this computational testbed, we run simulations of experiments, develop testable hypotheses, and then verify these hypotheses with human subjects in the lab.

We focus on repeated coordination games, because by definition they must have multiple equilibria. Because these games have multiple equilibria, many experiments have been run on these games to examine the equilibrium selection process. Therefore there are some well accepted behaviors which have been observed repeatedly in the experimental lab. In one game, the minimum-effort coordination game, the players have identical interests. In experiments with minimum-effort coordination games, Van Huyck, Battalio & Beil (1990) and Goeree & Holt (2005) show that subjects typically converge to a Nash equilibrium, but which Nash equilibrium depends on the number of subjects and the risk involved. In the other game, the battle of the sexes game, the players have opposing interests. Often in the battle of the sexes game, subjects learn to alternate between the two equilibrium outcomes as exhibited in Rapoport, Guyer & Gordon (1976) and McKelvey & Palfrey (2001).

The computational agents' decisions are made based on a learning algorithm. The major difference between this and previous learning algorithms is that we focus on "short" games that are repeated 100 rounds or less. Rather than developing an algorithm that has guaranteed long run convergence properties, we develop an algorithm that can generate data that resembles the data of human subjects in the experimental lab for these "short" games.

The task of developing computational agents could shed new light onto the theory of learning in games. The theory literature on learning in games has typically focused on learning algorithms with guaranteed convergence to some set, or small superset, of the Nash equilibria. The task of creating agents that mimic humans is different. By thinking of this new task, we may gain new insight into the decision making process in repeated games. For example, the algorithm proposed in this paper uses a pattern recognition scheme. Pattern recognition has been ignored in most of the theory literature, even though patterns have been observed in the experimental laboratory. Focusing on this new task may help uncover aspects that were previously missing in the theory.

This computational model of learning can also compliment experimental economics. We use computational agents to run simulations for both of these coordination games for a variety of experimental parameters. From these simulations, we develop hypotheses on human behavior which are then tested in the lab. We can run the simulations without any hypothesis in mind, and then develop these testable hypotheses from the simulation data. Since this is an alternative to economic theory, it may provide new insights into the human decision making process that haven't been examined in the theoretical literature.

The rest of the paper proceeds as follows. Section 2 discusses the related work in this area. Section 3 lays out the model. This includes the setup of the games and equilibria, as well as a detailed description of the learning algorithm. Section 4 compares experimental data to simulation

data from my algorithm as well as other learning algorithms. It also includes simulation results for both games which are used to develop testable hypotheses for experiments. Section 5 gives the experimental design as well as a comparison of the experimental and simulation results. We provide concluding remarks in Section 6.

2 Related Work

The theoretical literature on learning in repeated games started with the introduction of fictitious play by Brown (1951). In each round of fictitious play, each player best responds to the average of past plays. Fictitious play has been shown to converge to Nash equilibria beliefs for two player zero-sum games (Brown 1951), 2×2 games (Miyasawa 1961), potential games (Monderer & Shapley 1996), and many classes of supermodular games (Milgrom & Roberts 1991, Hahn 2008, Berger 2007). Fictitious play has garnered much interest from game theorists because of its simplicity and its nice convergence properties. However, fictitious play does have desirable equilibrium selection properties.

Another common approach is to use evolutionary game theory as introduced with the concept of the evolutionary stable strategies in Smith & Price (1973). In these models, strategies with random mutations compete against each other based on a fitness measure. Using an evolutionary model, Friedman (1991) shows that ESS are a subset of Nash equilibria. Adding randomness to the evolutionary model, Kandori, Mailath & Rob (1993) and Young (1993) show convergence to risk dominant equilibrium in 2×2 symmetric games. Hart (2002) introduces a model in which the unique ESS leads to the subgame perfect Nash equilibrium. These evolutionary models provide stronger equilibrium selection properties than fictitious play, but they typically take thousands of rounds to attain these equilibria.

Most of the literature examining interaction in repeated games has focused on long run convergence properties; a few papers develop computational agents which act like human subjects. Arifovic & Ledyard (2005) build computational agents to be used as a testbed for experiments on the Groves-Ledyard mechanism. In particular, the mechanism has one parameter which plays an important role in the speed of converge. They make predictions about optimal values of this parameter with their computational testbed, and then confirm these predictions with experiments. Their learning algorithm is a combination of a genetic algorithm with some behavioral intuition. Their computational agents are able to converge quickly, on average 20 rounds. However, their algorithm strongly favors convergence to a single choice. My algorithm is able to get quick convergence in some games, but in addition it also converges to move complex strategies in other games.

One of the key features of my algorithm is a pattern recognition scheme. Pattern recognition is an important aspect of human decision process. Sonsino & Sirota (2003) find that over half of the human subjects converge to patterns of Nash equilibria (where different NE are played in subsequent rounds) in a variation of the battle of the sexes game. The only way that these patterns of Nash equilibria can be sustained is if players are recognizing patterns. However, with the exception of a few papers, previous learning models have not allowed agents to recognize these patterns in repeated interactions.

Sonsino (1997) proposes a theoretical model which players recognize patterns, and play converges to a sequence of Nash equilibrium with probability one. This would include the alternations observed experimentally in the battle of the sexes. However, he does not address how these patterns arise, rather that the players are able to recognize them once they have been played. Also,

he does not propose a specific mechanism from which the players are able to recognize patterns. In another theoretical paper, Lambson & Probst (2004) create a model where players find similar patterns in the history, and best respond to these patterns. Their learning algorithm converges to the convex hull of the set of Nash equilibria in games where fictitious play converges to the set of Nash equilibria. This paper does not give a specific technique for determining the length of patterns that players will recognize. Rather, they assume that players only recognize patterns of a fixed length, and then examine the convergence of these players. Unlike these theoretical pattern recognition models, we actually implement computational agents which have a specific mechanism to recognize patterns of different lengths and select the optimal pattern lengths. Therefore, these agents are able to converge to single Nash equilibria as well as more complex patterns of Nash equilibria.

3 Model

3.1 Game

Before introducing the algorithm, we introduce the general game. The algorithm works on any variation of this general game. The game consists of n players $N = \{1, 2, \dots, n\}$. The set of strategy profiles is $S = S_1 \times S_2 \times \dots \times S_n$, where the player's action comes from the infinite strategy space $s_i \in S_i = [0, 1]$. A specific strategy profile is the set of all player's actions, denoted by s . Let s_{-i} denote the actions of all players other than player i in strategy profile s . Each player has a payoff function which maps strategy profiles to real numbers, $\pi_i : S \rightarrow \mathbb{R}$.

The analysis in this paper focuses on two specific games from the general class of games described above: the minimum-effort coordination game and the battle of the sexes game. Both of these games are generalized to the continuous action space. The minimum-effort coordination game has n players, $N = \{1, \dots, n\}$, each of which exerts a costly effort $s_i \in S_i = [0, 1]$. The payoff for player i is a term proportional to the minimum effort of all players, minus the cost of their own efforts. More formally,

$$\pi_i(s_i, s_{-i}) = \min_{j \in N} s_j - cs_i$$

for some cost parameter c . The continuous version of the battle of the sexes game has two players, $N = \{1, 2\}$. Each player chooses a location on the unit interval, $s_i \in [0, 1]$. Based on the player's locations, the payoff function is,

$$\begin{aligned}\pi_1(s_1, s_2) &= (2s_1 - 1 - b)(2s_2 - 1 - b) \\ \pi_2(s_1, s_2) &= (2s_1 - 1 + b)(2s_2 - 1 + b)\end{aligned}$$

for some parameter b . To get a better understanding of the battle of the sexes payoff functions, consider the two by two game where the payoffs are the same, but the actions space is now restricted to two choices, $S_i = \{0, 1\}$. The payoff matrices for different values of b are shown in Figure 1. There are a few things to notice about variations in b . First, it is symmetric around 0, o.e. the game is the same for $b = 0.1$ and $b = -0.1$ except that the players roles are switched. Without loss of generality we will only consider $b \geq 0$. When b is close to zero, the payoff difference between the equilibria at $(0, 0)$ and $(1, 1)$ is small while the cost of not coordinating (playing $(0, 1)$ or $(1, 0)$) is high. When b is close to 1, then the payoff difference between the two equilibria is large, but the cost of not coordinating is low. So this b parameter can be thought of as a symmetry parameter.

	0	1		0	1		0	1
0	(1, 1)	(-1, -1)	0	(2.25, .25)	(-.75, -.75)	0	(4, 0)	(0, 0)
1	(-1, -1)	(1, 1)	1	(-.75, -.75)	(.25, 2.25)	1	(0, 0)	(0, 4)
	$b = 0$			$b = .5$			$b = 1$	

Figure 1: Battle of the sexes game for different values of b .

3.2 Equilibria

3.2.1 Minimum-Effort Coordination Game

The set of Nash equilibria of the minimum-effort coordination game depends on the parameter c . If $c > 1$, then the cost of effort is greater than the benefit, so it is a dominant strategy for each player to play the minimum effort, $s_i = 0$. If $c < 0$, since the cost of effort is negative, it is a dominant strategy for each player to exert the maximum effort, $s_i = 1$. The case that is usually studied is $0 \leq c \leq 1$. In this case, a Nash equilibrium is any strategy where all players exert the same effort. The set of Nash equilibria is then defined by,

$$NE = \{s | s_i = s_j \text{ for all } i, j \in N\}$$

Since the strategy space is infinite, there are infinite Nash equilibria. These equilibria are Pareto ranked. The equilibrium where all players exert the lowest effort is the Pareto-worst equilibrium, as each player gets payoff $\pi_i(0, 0, \dots, 0) = 0$. The Pareto-optimal equilibrium is the equilibrium in which each player exerts the maximum effort and gets payoff $\pi_i(1, 1, \dots, 1) = 1 - c$.

A player can always guarantee a payoff 0 by playing 0, so the minmax payoff in this game is 0 for all players. By the folk theorem for repeated games, this implies that any strategy that guarantees the players at least 0 is an equilibrium in the infinitely repeated game.

3.2.2 Battle of the Sexes Game

In the continuous battle of the sexes game, when $b > 1$, each player has a strictly dominant strategy, so there is a unique equilibrium. The more interesting case occurs when $0 \leq b \leq 1$, in which there are three equilibria. Given that player 2 is located at s_2 , then the best response function for player 1 is as follows,

$$BR_1(s_2) = \begin{cases} 0 & s_2 < \frac{1+b}{2} \\ [0, 1] & s_2 = \frac{1+b}{2} \\ 1 & s_2 > \frac{1+b}{2} \end{cases}$$

Given that player 1 is located at s_1 , then the best response function for player 2 is,

$$BR_2(s_1) = \begin{cases} 0 & s_1 \leq \frac{1-b}{2} \\ [0, 1] & s_1 = \frac{1-b}{2} \\ 1 & s_1 > \frac{1-b}{2} \end{cases}$$

Therefore, the three Nash equilibria are $(s_1, s_2) = (0, 0), (\frac{1-b}{2}, \frac{1+b}{2}), (1, 1)$. The equilibrium $(s_1, s_2) = (\frac{1-b}{2}, \frac{1+b}{2})$ is Pareto-dominated by the other two equilibria. However, the other two equilibria are not Pareto rankable. Player 1 prefers the equilibrium $(s_1, s_2) = (0, 0)$ and player 2 prefers the equilibrium $(s_1, s_2) = (1, 1)$.

In this game, each player has a strategy which guarantees a payoff of 0, and this turns out to be the minmax payoff. By playing $s_1 = \frac{1-b}{2}$, player 1 will get a payoff of zero regardless of what player 2 plays. Similarly, player 2 can guarantee zero by playing $s_2 = \frac{1+b}{2}$. By the folk theorem, this implies that any strategy where the players are guaranteed a positive average payoff is an equilibrium of the infinitely repeated game. This includes the strategy where the players alternate between the two endpoints, which gives an average payoff of $\frac{1}{2} \left[(1-b)^2 + (1+b)^2 \right]$ per round.

3.3 Algorithm

In each period of a repeated game, the algorithm determines which choice each agent makes. This choice depends on the history of play as well as the current state of the agent. After each agent has made their choice, the choices and payoffs are revealed to all agents. The agents then update their history and current state, and make their choice for the following round.

Two main features of this algorithm are the pattern recognition scheme and the agent's states. The experiments of Sonsino & Sirota (2003) show that human subjects are able to sustain non-trivial patterns of Nash equilibria. Even in 2-by-2 games, the probability of sustaining a pattern of Nash equilibria for n rounds by random choice decreases exponentially as n increases, yet the subjects are still able to sustain these patterns. The human subjects' ability to sustain these patterns of equilibria provide evidence that they are in fact recognizing these patterns. Therefore, pattern recognition is a natural feature when modeling human behavior in repeated interactions. My pattern recognition scheme is a modification of the k-nearest neighbor classification algorithm from machine learning. Patterns are recognized by first identifying the current play (the most recent choices in the history) and then finding previous plays that are similar to the current play. The prediction for next round is a weighted average of the outcomes of these similar plays. In each round, the agent makes their choice based on their current state, which is given by two parameters, γ and σ . The γ parameter represents the current level of confidence for an agent. This is determined by how well that agent is predicting what the other agents will do. The σ parameter represents the agent's satisfaction of the current play of the game. If the agent is not satisfied, and wants to change what is happening in the game, then σ is close to 1. If the agent is satisfied with how the game is going then σ is close to 0. When all agents have high values of γ and low values of σ , then each agent's choice has low variance and each agent is satisfied with the predicted outcome of their choice, so the algorithm has converged.

Another important aspect in the algorithm is that the agents are not able to calculate exact best-responses to their predictions. Instead, the agents determine best responses by randomly sampling from the strategy space, and keeping the strategy that gives the highest payoff. This is important for two reasons. First, it allows for completely general payoff functions. Since the explicit best response function isn't required, the payoff functions need not be continuous or differentiable. Also, it allows agents to have different levels of intelligence by changing the number of samples they take. For example, a very intelligent agent has a good grasp of the payoff function, and therefore is able to find the best response. This can be modeled by an agent who takes a large number of random samples to find the best response. Conversely, a very unintelligent agent is not able to find

the best response. This can be modeled as an agent that takes a very small number of samples to find the best response.

The explanation of the algorithm is divided up into four parts: notation, preliminary initialization, round k action, and preparation for round $k + 1$. For notational purposes, the superscript typically denotes the agent and the subscript denotes the round.

3.3.1 Notation

Each agent has a database of information that is used to help make their choice in each round. At the start of each round, each agent has two parameters in their database, the confidence parameter γ and the satisfaction parameter σ . These parameters for agent i in round k are denoted by γ_k^i and σ_k^i . The agents use these parameters to help make their choice. Agent i 's choice in round k is represented by $\mathbf{x}_k(i)$. The choice of all agents in round k is given by \mathbf{x}_k , which yields payoffs $\pi_i(\mathbf{x}_k) = \pi_k^i$ for agent i .

After the agent makes their choice, they update their database of information in preparation for the next round. Each agent makes a prediction about what the other agents will play in the following round. Let $\hat{\mathbf{x}}_k^i(j)$ be agent i 's prediction for agent j 's play in round k . The full prediction vector, $\hat{\mathbf{x}}_k^i$, consists of predictions for all of the other agents.

As the game progresses, each agent creates a quasi-best-response matrix. Agent i 's quasi-best response matrix at round k is denoted by Q_k^i . This matrix helps the agent determine what they should choose after they have made their prediction. To do this, the agent groups similar strategy profiles together in the quasi-best-response matrix. The agent then determines which play is best against these similar strategy profiles by randomly sampling responses from the strategy space. In the future, when a similar strategy profile arises, the agent uses this quasi-best-response matrix to help remember what they did in the past. From this quasi-best-response matrix, the agent determines the quasi-best-response for their prediction for round k , which is denoted by x_k^{i*} . More details about the quasi-best-response are given below in the description of the algorithm in the preparation for round $k + 1$ section.

Each agent also keeps track of their best and worst outcomes. To do this, each agent randomly chooses J strategy profiles from the uniform distribution on the joint strategy space $S = [0, 1]^N$. Next, they calculate the payoffs for each of these profiles, and save the strategy profiles which yield the highest and lowest payoffs, $\bar{\mathbf{x}}_k^i$ and $\underline{\mathbf{x}}_k^i$ respectively. These are referred to as the highest and lowest known choices for agent i in round k . The payoffs for these strategy profiles, $\bar{\pi}_k^i$ and $\underline{\pi}_k^i$, are referred to as the highest and lowest known payoffs for agent i at round k .

All of this information is stored in the agent's database, and is available when they are making their choice in round k .

3.3.2 Initialization

Many learning algorithms contain multiple initialization periods, where the agents choose randomly in the strategy space. Since the focus of this paper is not long run convergence, but rather short run behavior, the initialization period has to be short. Before the first choice is made, the agents randomly choose J strategy profiles to determine their initial highest and lowest known payoffs, $\bar{\mathbf{x}}_0^i$ and $\underline{\mathbf{x}}_0^i$ respectively. Each agent then makes the initial predictions about the other agents by randomly drawing a number from the uniform distribution on $[0, 1]$, that is $\mathbf{x}_1^i(j) \sim U[0, 1]$. Finally, each agent starts with the lowest possible confidence level, $\gamma_1^i = 10$. They also start with the highest

satisfaction parameter, $\sigma_1^i = 0$, because they have no reason to try to change the outcome of the game yet. With these initial parameters, the algorithm is ready to run.

3.3.3 Round k

Entering round k , agent i has a database of information which is used to make a choice in round k . The choice in round k is a random number from a beta distribution with mean μ and variance ν^2 . The mean of the distribution is a convex combination of the quasi-best-response, x_k^{i*} , and the strategy which yields highest known payoff for agent i at round k , \bar{x}_k^i . The weight on each term is determined by the current level of satisfaction. If the agent's satisfaction level is high ($\sigma_k^i = 1$) then they play the quasi-best-response for their prediction. If the agent is not satisfied ($\sigma_k^i < 1$), then they try to move the outcome towards the point which yields their highest known payoff. That is,

$$\mu = \sigma_k^i x_k^{i*} + (1 - \sigma_k^i) \bar{x}_k^i$$

The variance of the distribution is inversely proportional to the current level of confidence¹. The proportionality constant is ρ , so the variance is,

$$\nu^2 = \frac{1}{\rho \gamma_k^i}$$

As the confidence level of the agent increases, the choice distribution has lower variance, and therefore the choice is more accurate. When the agent is not confident about what the other agents will do, then his choice distribution has high variance, and his choice is not as accurate.

After all agents have made their choices as described above, the payoffs are calculated. The agents then learn the choices of the other agents as well as the payoffs of all agents. At this point, the agents begin their preparation for round $k + 1$ by updating their database of information.

3.3.4 Preparation for Round $k + 1$

The agents have a variety of tasks to perform in preparation for round $k + 1$.

Update extremes As the game progresses the agents become more acquainted with the payoff function. To model this, each round the agents update their highest and lowest known payoffs by taking J random samples from the joint strategy space. For each random sample \mathbf{z}_j , the payoff vector is calculated. If the payoff for agent i from the sample is higher than the highest known payoff for agent i in round k , then the agent sets the highest known choice for round $k + 1$ to $\bar{\mathbf{x}}_{k+1}^i = \mathbf{z}_j$ and the highest known payoff for round $k + 1$ to $\bar{\pi}_{k+1}^i = \pi_i(\mathbf{z}_j)$. If none of the payoffs from the J sample points are higher than the highest known payoff for agent i at round k , then the highest known choice and payoff from round k are carried over to round $k + 1$, i.e. $\bar{\mathbf{x}}_{k+1}^i = \bar{\mathbf{x}}_k^i$ and $\bar{\pi}_{k+1}^i = \bar{\pi}_k^i$. The same update is performed for the lowest known play and payoff.

¹It is not possible to have a distribution over a closed region, if the variance is high, and the mean is sufficiently close to the endpoints. If this is the case, then it is corrected by using a modified beta distribution with mass point on the endpoint.

Prediction for round $k + 1$ In order to make a choice in round $k + 1$ it is useful for the agents to have some prediction about what their opponents are going to do in round $k + 1$. The prediction scheme used by the agents is a modification of the nearest neighbor classification algorithm from machine learning. The goal of the prediction scheme is to make a prediction for \mathbf{x}_{k+1} . Since there are N agents, the agents choices at round k are given by the vector $\mathbf{x}_k \in \mathbb{R}^N$. A pattern is vector combining one or more of these choice vectors. For example, a pattern of length 3 is $[\mathbf{x}_k \ \mathbf{x}_{k+1} \ \mathbf{x}_{k+2}]$. The agents divide the history of choice into the current pattern, previous patterns, and outcomes. Each previous pattern has a corresponding outcome. The algorithm makes a prediction for the outcome of the current pattern. The agents determine which of the previous patterns are closest to the current pattern. Then the agents prediction is a weighted sum of the outcomes of the closest patterns. The agents repeat this process for patterns of different lengths, n . After the agent has done this for all values of n , he compares them, and determines which pattern length provides the best prediction.

For example, consider a two player game with the history of play after eight rounds,

$$(0,0), (1,1), (1,1), (0,0), (1,1), (1,1), (0,0), (1,1)$$

Let's examine the prediction by agent 1 of what agent 2 will play in the ninth round. First, agent 1 considers patterns of length 1. The current pattern is the most recent play, $(1,1)$. This has been played four previous times in rounds 2,3,5 and 6. These are the closest patterns. When these closest patterns have been played in the past, agent 2 has responded by playing 1,0,1, and 0 in the respective following rounds. These are the outcomes for the four closest patterns. This is not good, because agent 2 has played 0 half the time, and 1 half the time, so it is difficult to predict what agent 2 will play in the next round based on patterns of length 1.

Next, agent 1 looks at patterns of length 2. The current pattern in this case is the play in the previous 2 rounds, $(0,0), (1,1)$. This pattern has been played twice before in the past, in rounds 1-2 and 4-5. In response to this pattern, agent 2 has played 1 in both rounds 3 and 6. After patterns of length 2, agent 2 always chose 1. Therefore, patterns of length 2 are better for prediction than patterns of length 1.

More formally, at the end of the k^{th} round, each agent considers patterns of different lengths n . For each length, there are $k - n$ previous patterns of length nN each. The agent forms the previous patterns matrix $X \in \mathbb{R}^{k-n \times nN}$ and the output matrix $Y \in \mathbb{R}^{k-n \times N}$,

$$X = \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_n \\ \mathbf{x}_2 & \cdots & \mathbf{x}_{n+1} \\ \vdots & & \vdots \\ \mathbf{x}_{k-n} & \cdots & \mathbf{x}_{k-1} \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_2 \end{bmatrix} \text{ and } Y = \begin{bmatrix} \mathbf{x}_{n+1} \\ \mathbf{x}_{n+2} \\ \vdots \\ \mathbf{x}_k \end{bmatrix}$$

Each row of the previous patterns matrix is a single pattern, and these are denoted by X_m for $m = 1, \dots, k - n$. The current prediction is the vector $c \in \mathbb{R}^{nN}$

$$c = [\mathbf{x}_{k-n+1} \ \cdots \ \mathbf{x}_k]$$

Next, the agent finds the j rows of X which are closest to the current pattern c in terms of Euclidean distance. To do this, the agent forms the distance vector by finding the length between

the current point and each of the previous points,

$$\mathbf{d} = \begin{bmatrix} \|X_1 - c\| \\ \|X_2 - c\| \\ \vdots \\ \|X_{k-n} - c\| \end{bmatrix}$$

Let J be the set of indices of the j smallest terms in the distance vector \mathbf{d} . That is $d_j \leq d_k$ for $j \in J$ and $k \notin J$. These indices correspond to the j rows of X which are closest to the current point c .

The agent now determines which pattern length gives the best prediction. As exhibited in the above example, the agent wants to choose the pattern length with the most similar outcomes. To determine the optimal pattern length for each n , the agent takes the outcome of the j closest points, and calculates the average of these points, \bar{Y} . Then the agent computes the variance of these j closest points,

$$V_n = \sum_{j \in J} \|Y_j - \bar{Y}\|$$

Now, the agent compares the variance for all considered pattern lengths and chooses the pattern length with the smallest variance. If there is a tie, then the agent chooses the shorter pattern. Note that average variances are higher in higher dimensions. This is not corrected for, which gives an additional benefit to the shorter patterns, because shorter patterns are easier to recognize.

Once the agent has selected which pattern length to use, he forms a weighted average of the closest outcomes. The closer the pattern is to the current outcome, the higher the weight is. The patterns are weighted using a logistic function. The prediction for the next period is thus,

$$\hat{\mathbf{x}}_{k+1}^i = \frac{\sum_{j \in J} Y_j e^{\mathbf{d}(j)}}{\sum_{j \in J} e^{\mathbf{d}(j)}}$$

Therefore, if the distance to each of the j closest patterns is 0, then the prediction is just the average outcome from those j closest patterns. The agent makes their choice for period $k+1$ based on this prediction.

Quasi-Best-Response The quasi-best-response helps the agent determine the best response for his prediction for round $k+1$. To do this the agent updates the quasi-best-response matrix from the previous period, Q_k^i . Each row of the quasi-best-response matrix consists of three items: prediction about what the other agents will do, what agent i should do given that prediction, and the payoff given that strategy profile. More formally row m has the terms,

$$Q_k^i = [\mathbf{q}_{-i}^m \quad \mathbf{q}_i^m \quad \pi_i(\mathbf{q}_i^m, \mathbf{q}_{-i}^m)]$$

Here, \mathbf{q}_{-i}^m are the choices of the other agents, and \mathbf{q}_i^m is the choice of agent i . Agent i updates Q_k^i as follows. First, agent i determines if the current prediction is similar to any of the entries already in the quasi-best-response matrix. To do this, agent i chooses a set, R , of random strategies. For each row of the quasi-best-response matrix, agent i calculates the payoff difference,

$$pd_m = \sum_{r \in R} |\pi_i(r, \mathbf{q}_{-i}^m) - \pi_i(r, \hat{\mathbf{x}}_{k+1}^i)|$$

Next, the agents find the minimum payoff distance, $pd^* = \min pd_m$. If the distance is small, i.e. $pd^* < \delta$, then the two strategies are similar, and therefore are combined in the quasi-best-response matrix. If $pd^* > \delta$, then the two strategies are not similar, so a new entry is created in the quasi-best-response matrix. Let the threshold δ be a fraction of the difference between the highest and lowest payoff,

$$\delta = \frac{\bar{\pi}_i^k - \underline{\pi}_i^k}{20}$$

If $pd^* < \delta$, then the agent updates the row of the quasi-best-response matrix corresponding to pd^* , call this row m^* . The agent takes the set of R strategies, and calculates the payoffs $\pi_i(r, \hat{\mathbf{x}}_i^{k+1})$. Let r^* denote the strategy from R which maximizes $\pi_i(r, \hat{\mathbf{x}}_i^{k+1})$. If this new strategy yields a higher payoff than the current quasi-best-response, i.e. $\pi_i(r^*, \hat{\mathbf{x}}_i^{k+1}) > \pi_i(\mathbf{q}_i^m, \mathbf{q}_{-i}^m)$, then the row m^* is updated to $\mathbf{q}_{-i}^{m^*} = \hat{\mathbf{x}}_i^{k+1}$ and $\mathbf{q}_i^{m^*} = r^*$.

If $pd^* > \delta$, then the agent creates a new row for the quasi-best-response matrix, call this row $M+1$. Again, the agent calculates the payoffs $\pi_i(r, \hat{\mathbf{x}}_i^{k+1})$ for all $r \in R$, with r^* being the strategy which yields the maximum payoff. The agent then updates the quasi-best-response matrix by setting $\mathbf{q}_{-i}^{M+1} = \hat{\mathbf{x}}_i^{k+1}$ and $\mathbf{q}_i^{M+1} = r^*$.

Update γ The parameter γ measures the current level of confidence of the agent. When the agent makes accurate predictions, his confidence increases. In preparation for round $k+1$, the agent compares his prediction for round k that was made in round $k-1$, $\hat{\mathbf{x}}_k^i$, with the actual play from round k , \mathbf{x}_k . Based on this prediction and outcome, the agent updates his confidence as follows,

$$\gamma_{k+1} = \frac{\alpha_1}{\|\hat{\mathbf{x}}_k^i - \mathbf{x}_k\| + \alpha_2} \gamma_k$$

Therefore, if the Euclidean distance between the prediction and the actual outcome is less than $\alpha_1 - \alpha_2$, then the confidence increases. The maximum possible increase in confidence is α_1/α_2 .

Update σ The parameter σ represents the agent's satisfaction at the current state of the game. If the agent is not satisfied with the current outcome, then he may try to induce the other agents to play something else in order to change the current outcome. If the agent's attempt to move is unsuccessful, then he will stop trying. For example, suppose two agents are coordinating at one of the equilibria repeatedly in the battle of the sexes game. Agent 1 is at her optimal equilibrium, and Agent 2 is at his least favored equilibrium. Agent 2 realizes that he can receive a higher payoff at the other equilibrium. Therefore he will try to induce agent 1 to start playing the other equilibrium. However, agent 1 may not change the way she is playing, even when agent 2 starts playing something else. If agent 2 has tried for a long time with no success, he will give up, and start playing the original equilibrium. The entire process of trying to move and giving up is called a *moving session*.

Agent i will start with the highest satisfaction possible. The satisfaction will remain at the highest level until some event causes agent i to start a moving session. In order for the agent to become dissatisfied, he has to have a good idea of what the other agents are going to play. Therefore, agent i must have a confidence greater than γ_{MS} in order to start a moving session. Given that agent i has confidence greater than γ_{MS} , he will start a moving session in two situations.

If agent i knows that all agents receive higher payoffs at his highest known play, then he will try to move there because everyone will receive a higher payoff. Also if agent i 's highest known payoff increases agent i 's payoff by a large amount, and decreases the other agents' payoffs by only a small amount, then he will try to change the outcome. There are also some situations in which agent i will not start a moving session, even if his confidence is greater than γ_{MS} . If moving to agent i 's highest known play will increase agent i 's payoff by a small amount, but will decrease all other agents payoffs by a large amount, then agent i will not try to change the outcome. Also, if agent i has tried to move before unsuccessfully, then he will not try to move again until he has found a better strategy.

Once the moving session has started, agent i will try to induce the other agents to play his optimal strategy. If the play of the game is moving away from the play at the start of the moving session, and towards the highest play for agent i , then agent i will continue the moving session. If the play of the game does not move towards the highest play for agent i , then that round will be considered a *failure*. If the total number of failures become to high, then i will stop the moving session.

To more formally define this event that triggers a moving session, consider the term,

$$\Sigma_k^i = \frac{\pi_i(\bar{\mathbf{x}}_k^i) - \pi_i(\mathbf{x}_k)}{\frac{1}{N-1} \sum_{j \neq i} \pi_j(\bar{\mathbf{x}}_k^i) - \pi_j(\mathbf{x}_k)}$$

Σ_k^i will be referred to as the *relative gain* for agent i in round k . Agent i 's payoff at the highest known play is always greater than his payoff at the current play, because the agent takes the current play into account when updating his highest known play. Therefore, switching from the current play \mathbf{x}_k to agent i 's highest known play $\bar{\mathbf{x}}_k^i$ will always increase agent i 's payoff. So the numerator of Σ_k^i will always be positive.

The agent will also keep track of the *maximum relative gain* for round k , $\bar{\Sigma}_k^i$, and the *minimum relative gain* for round k , $\underline{\Sigma}_k^i$. At the beginning of the game, agent i will start with maximum relative gain of $\bar{\Sigma}_0^i = 0$ and minimum relative gain of $\underline{\Sigma}_0^i = -1$. The agent will update these extreme relative gains with the current relative gain when the current relative gain is more extreme (higher than maximum or lower than minimum) and confidence is greater than γ_{MS} . The role of the extreme relative gains is to ensure that the agent does not continuously try to move to a point which the other agents refuse to move to.

Based on the current relative gain, the extremes relative gains, and the confidence, agent i will determine whether or not to start a moving session. When the denominator of Σ_k^i is positive, and hence $\Sigma_k^i > 0$, the other agents will benefit on average when switching from \mathbf{x}_k to $\bar{\mathbf{x}}_k^i$. So, if $\Sigma_k^i > \bar{\Sigma}_k^i$ and $\gamma_k^i > \gamma_{MS}$, then the agent will start a moving session because all agents will have higher payoffs at $\bar{\mathbf{x}}_k^i$. When the denominator of Σ_k^i is negative, the other agents will get lower payoffs on average when switching from \mathbf{x}_k to $\bar{\mathbf{x}}_k^i$. However, if Σ_k^i is very negative, then the average decrease of the other agents payoff will be small compared to the increase for agent i . So if $\Sigma_k^i < \underline{\Sigma}_k^i$ and $\gamma_k^i > \gamma_{MS}$ then the agent will also start a moving session. To summarize, agent i will try to move if $\Sigma_k^i \notin [\underline{\Sigma}_k^i, \bar{\Sigma}_k^i]$ and $\gamma_k^i > \gamma_{MS}$.

In the first round of the moving session, agent i will decrease from the full satisfaction level $\sigma = 1$ to the level $\sigma = \sigma_0 < 1$. Agent i will also set the number of failures to 0, $f = 0$. Agent i should not expect the other agents to respond to this move until they have seen the play in second round of the moving session and had at chance to respond to it in the third round of the moving session. So the agent will remain with satisfaction $\sigma = \sigma_0$ in the second round of the moving

session, and this will not count as a failure. Starting in the third round, agent i 's satisfaction and failures will depend on whether the other agents are responding to agent i 's move. In particular, if the other agents are responding, and play is moving toward the highest known payoff, i.e.,

$$\|\mathbf{x}_k - \hat{\mathbf{x}}_k^i\| > \|\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^i\|$$

then the satisfaction will increase, $\sigma_{k+1} = \bar{\xi}\sigma_k$ and the number of failures will stay constant $f_{k+1} = f_k$ (for some $\bar{\xi} > 1$). Alternatively, if the other agents are not responding, so play is not moving toward agent i 's highest known payoff, i.e.

$$\|\mathbf{x}_k - \hat{\mathbf{x}}_k^i\| < \|\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^i\|$$

then the satisfaction will decrease, $\sigma_{k+1} = \underline{\xi}\sigma_k$ and the number of failures will increase by one, $f_{k+1} = f_k + 1$.

When the amount of failures reaches the threshold $f_k = \bar{f}$, then the session ends because the other agents aren't responding to the move. After the session ends, the amount of failures is set back to 0, and the satisfaction is set back to the highest level $\sigma = 1$.

4 Simulation Results

To test the algorithm, we first compare simulations to some experimental results reported in the literature. We then compare the simulations of my algorithm to simulations using fictitious play and the algorithm proposed in Arifovic & Ledyard (2005). From now, fictitious play will be referred to as FP, Arifovic & Ledyard's (2005) algorithm as AL, and my algorithm as PR (pattern recognition).

4.1 Comparisons with Experiments

4.1.1 Minimum Effort Coordination Game

There have been several papers that run experiments with minimum effort coordinations games with continuous strategy spaces. Goeree & Holt (2005) (GH from now on) run experiments on the continuous minimum effort coordination game with group size $N = 2$. Each of their six sessions involved ten subjects which played a game for ten rounds. At the beginning of each round, each subject was randomly paired with another subject. Subjects were then asked to pick an effort level from the range $[110, 170]$. After all subjects had picked their effort, their payoffs and their partner's choice were revealed. They ran two treatments of three sessions apiece; one with low cost ($c = 0.25$), and another with high cost ($c = 0.75$). Their results are displayed in Figure 2. These experiments show that the choice's of the subjects in the high cost treatments become progressively lower, and the choice's of the subjects in the low cost treatments become progressively higher. In addition, the authors point out that late in the experiment, the choices in different treatments are separated by the mean of the range (140). GH also run similar treatments for 20 rounds, and find that the adjustment of choices tends to happen in the first 10 rounds, and the play levels out after that. These are the characteristics that are desired in the simulations using the PR algorithm.

The minimum effort coordination game experiments run in GH can be simulated with computational agents using PR to make their decision. One difference between the experiment and my setup is that GH use different random matched pairs each round rather than fixed pairs throughout the experiment. To correct for this, we set the satisfaction parameter $\sigma = 0$. The satisfaction

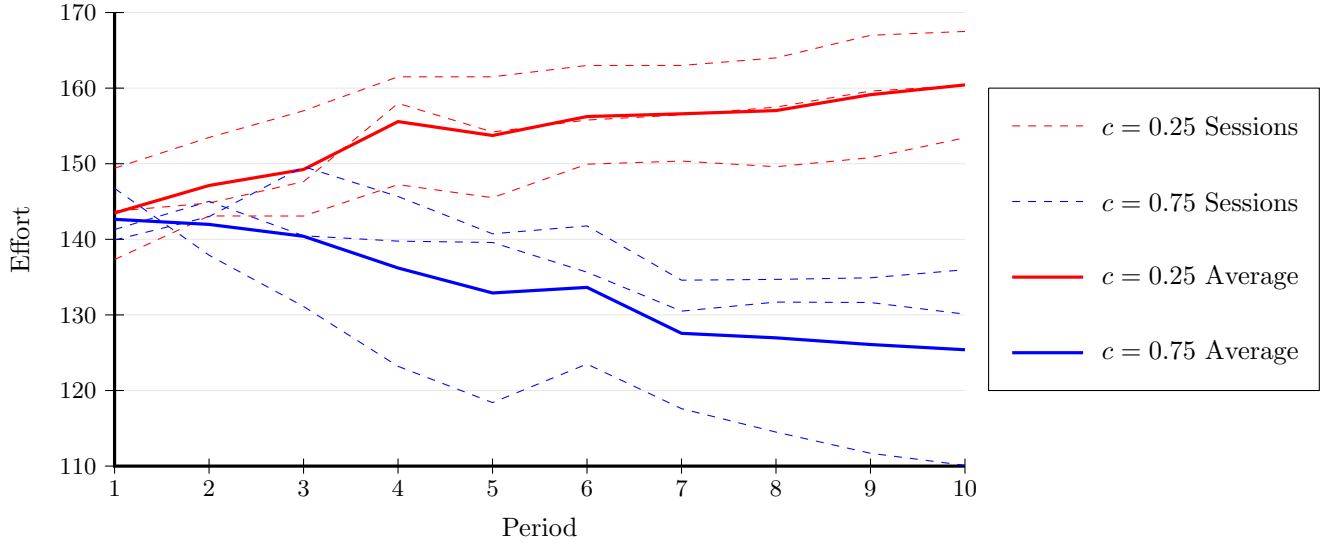


Figure 2: Experimental results of continuous minimum effort coordination game from Goeree & Holt (2005).

parameter allows the agents to induce the other agents to play some strategy. This parameter is set to zero because it is difficult for one agent to induce another agent to play something when they will both be randomly matched with another agent in the following round. The results from the simulations are presented in Figure 3. As in the actual experiment, there is separation between the high and the low cost treatments. In addition, the separation of the two treatments occurs around the midpoint as observed in the experiments. Finally, most of the adjustment happens in the first 10 rounds, and play levels out after that. So the agents in the simulations exhibit similar dynamics and equilibrium selection properties to the human subjects in the continuous minimum effort coordination game.

There have also been several papers which examine the effect of group size on coordination in the minimum effort coordination game. Van Huyck, Battalio & Beil (1990) (VHBB) run experiments with minimum effort coordination games with varying group size². They find that for large group sizes ($N=14-16$) with $c = 0.5$, subjects converge to the Pareto dominated equilibrium, where all players exert the lowest effort. They also find that after failing to coordinate on the Pareto superior equilibrium in large groups, these same subjects are able to converge to the Pareto superior equilibrium 12 out of 14 times when matched in fixed groups of two. So even when the subjects are used to playing the dominated equilibrium, they are still able to coordinate on the high effort when matched in fixed pairs. This provides strong evidence that fixed groups of two should be able to reach the Pareto superior equilibrium a large percentage of the time.

Knez & Camerer (1994) (CK) extend the result from VHBB by running similar experiments with more group sizes, and again using $c = 0.5$. They find that as the number of subjects increases, coordination on high effort strategies becomes more difficult. They also find that the biggest difference occurs when increasing the number of subjects from two subjects to three subjects. They

²Their subjects have 7 choices rather than a continuum of choices.

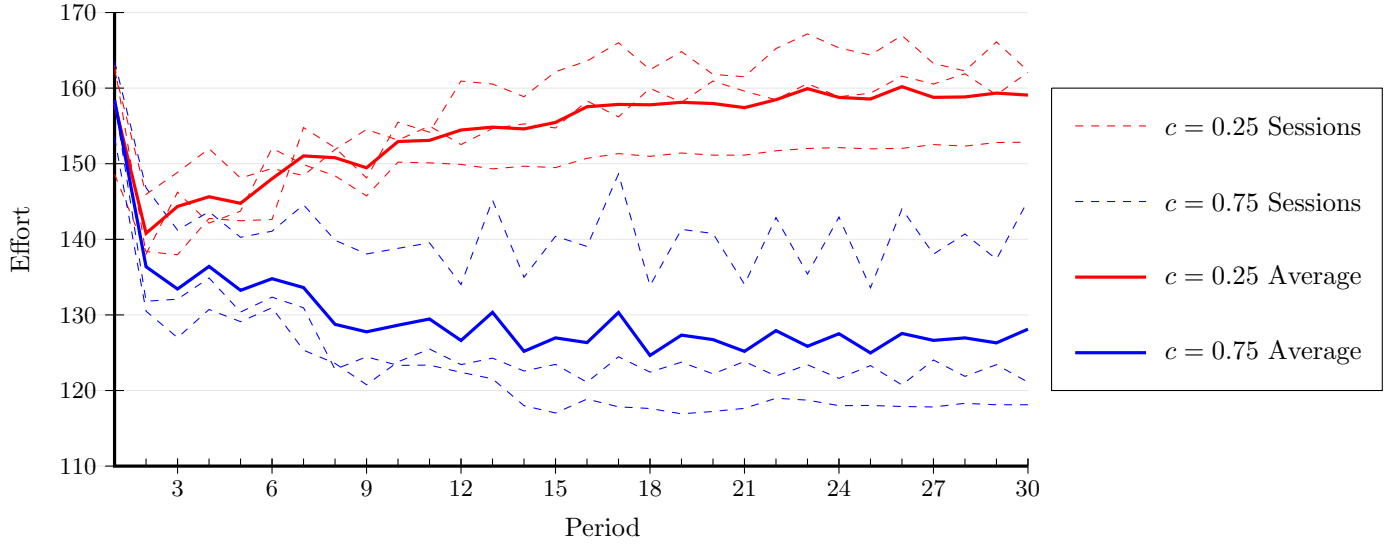


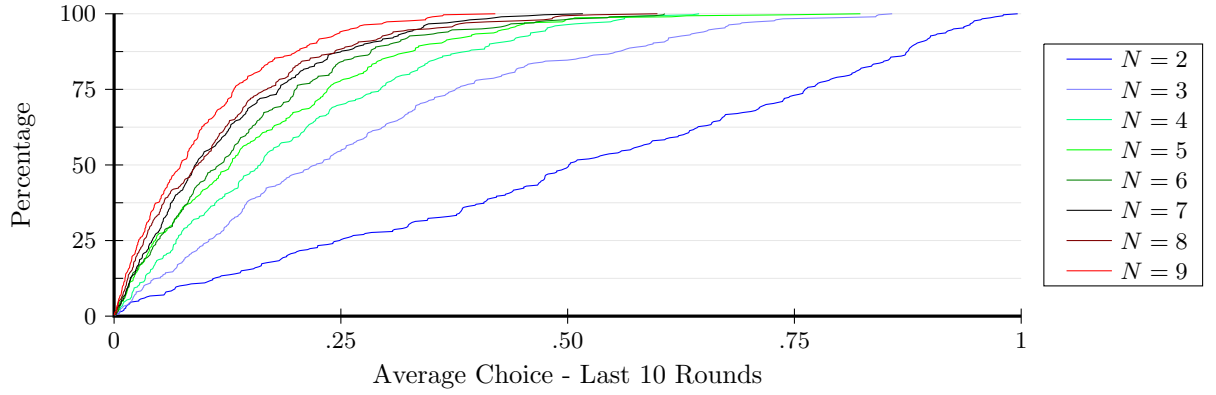
Figure 3: Simulation of continuous minimum effort coordination game experiments with agents using PR.

provide two reasons for this. First, picking an additional number from a random distribution will lead to a lower minimum. The second explanation is that the belief structure becomes ambiguous when increasing from two to three subjects, which causes players to more cautious when choosing their effort level.

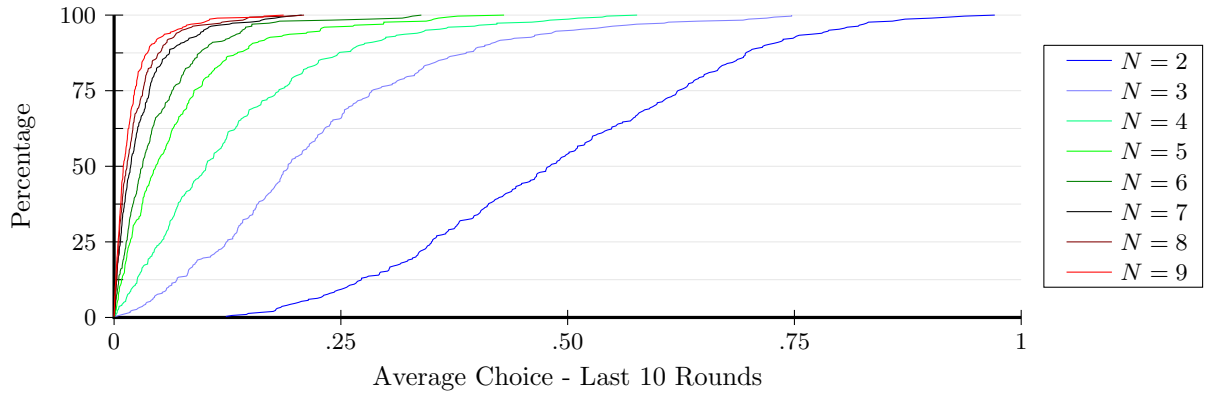
we run simulations of the minimum effort coordination game with different group sizes using FP, AL, and PR to see if they match the experimental results above. For each group size, we run 300 simulations of 100 rounds each with cost $c = 1/2$. We show that the simulations using PR are more similar to the experimental data than the simulations using FP and AL. The details of these simulations are left to the appendix.

To examine the outcomes of these simulations, we look at the average choice in the last 10 rounds. By the last 10 rounds, every agent was playing within 0.05 of the other agents over 99% of the time with FP, over 98% of the time with AL, and over 99% of the time with PR. So the average over the last 10 periods is a reliable measure of the convergence of strategies³. This average choice in the last 10 rounds is referred to as the convergence point of the simulation. The empirical cumulative distribution functions of convergence points for each of these learning algorithms are displayed in Figure 4. More coordination on the high effort outcomes is represented by ECDFs that are further to the southeast. The ECDF corresponding to perfect coordination on the high effort equilibrium would be 0% everywhere except 1. If the agent always converge on the low effort equilibrium, then the ECDF would be 100% everywhere except 0. The ECDFs show that the comparative statics on N match the experimental data for all three sets of simulations: as the number of agents increases, coordination on Pareto superior equilibria become more difficult.

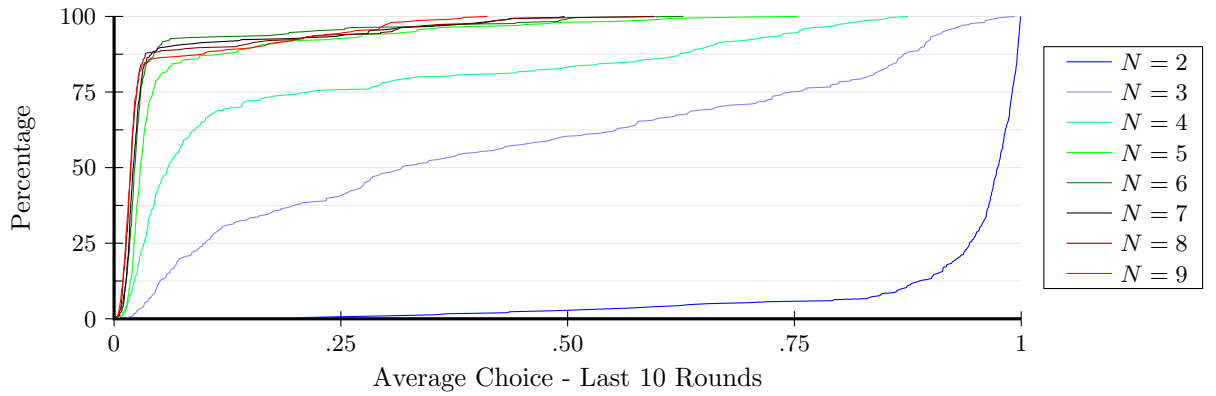
³The reason for using the average over the last 10 periods rather than the choice in the last period is to avoid problems where all of the agents have played the same thing for 9 rounds in a row, but in the last round, one agent changes what he plays.



(a) FP



(b) AL



(c) PR

Figure 4: Empirical cumulative distribution functions of convergence points for FP, AL, and PR for different group sizes.

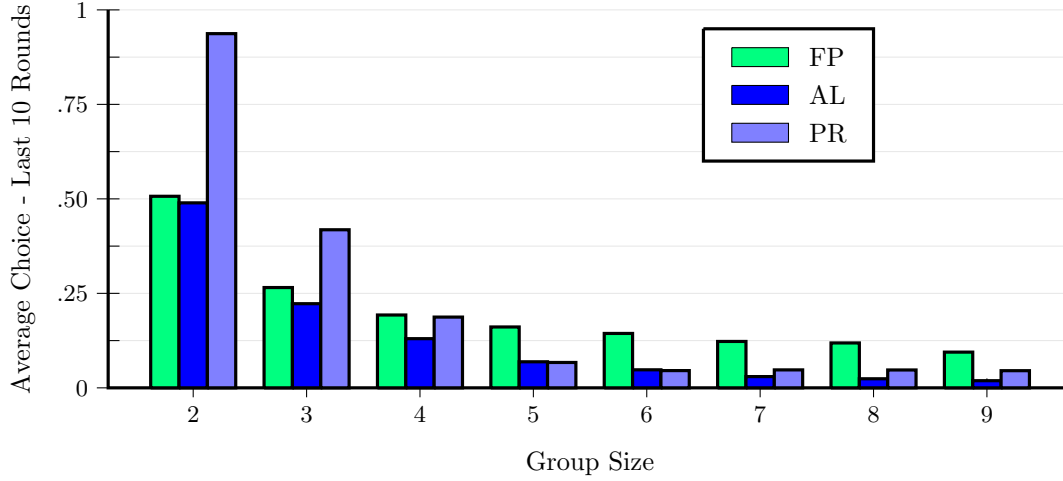


Figure 5: Average Convergence Points for FP, AL, and PR for different group sizes.

The average convergence point for each group size are summarized in Figure 5. The average convergence point is the expected outcome for a simulation, and corresponds to the mean of the ECDF for that simulation. As shown in CK, the figure shows that in all three simulations, the biggest difference in ability to attain the high effort outcome occurs when increasing group size from two to three. The FP and AL simulations have relatively low convergence points for groups of size two. The average convergence point for simulations with groups of two for FP is 0.51 and for AL is 0.49. This is in contrast to the experiments run in VHBB which show that fixed pairs of agents are able to converge to the Pareto superior equilibrium about 85% of the time. The average convergence point of the PR simulations is 0.94 which is more in line with the coordination behavior exhibited in VHBB by human subjects in the laboratory. Based on Figure 5, the PR algorithm matches the experimental data better than both AL and FP.

4.1.2 Battle of the Sexes Game

Next we run simulations of the battle of the sexes game using the three different algorithms. The battle of the sexes game, like the minimum effort coordination game, has multiple equilibria. But, in the battle of the sexes game, the equilibria are not Pareto ranked. One desirable outcome that has been observed in the literature is when players are able to alternate between the two equilibria. This maintains the maximum welfare, but also yields equal payoffs for both players. While most of the literature has focused on more complex extensions of the battle of the sexes game, this outcome has been observed in the simple battle of the sexes game as in Rapoport, Guyer & Gordon (1976) and more recently in Sonsino & Sirota (2003). To test to see if the learning algorithms can obtain this outcome, we run 300 simulations of 100 rounds each of the battle of sexes game for each learning algorithm. Each game consists of two players, and has cost $c = 0.5$.

When agents use FP to make their choice, two outcomes occur. Either the players converge to one of the equilibria, or they start a cycle of non-coordination. In the non-coordination cycle, the players will always choose the opposite location, and therefore it is not a Nash equilibrium and the

payoff is the lowest possible. Out of the 300 simulations, they reach this non-coordination cycle 41 times. Every other time they converge to one of the two equilibria. When agents use AL to make their choice, they have a tendency to converge to playing the same choice repeatedly. Therefore, it is difficult to converge to patterns of equilibria. In the simulations with AL, agents converge to one of the two equilibria every time out of 300 trials. So neither FP or AL converge to the pattern of Nash equilibria that has been observed experimentally.

Finally we run simulations where the agents use PR to make their choices. To determine if agents converge to a pattern of equilibria, we develop two types of convergence. First, there is convergence in strategies. Convergence in strategies occurs at the round when the difference between any agent’s choice in two consecutive rounds is less than 0.05 for the remainder of the game. This means the agents are playing the same thing repeatedly. The other type of convergence is convergence in γ , which happens when $\gamma > 5000$ for all agents for the remainder of the game. This means that all agents are making accurate predictions about what the other agents will do. Convergence to a pattern of Nash equilibria will occur when the agents converge in γ but not in strategies. Out of the 300 simulations when agents use PR to make their choice, they converge to a pattern of Nash equilibria 15 times. Out of these 15 patterns 14 consist of players playing one equilibrium in even rounds, and the other equilibrium in odd rounds. In the other pattern, agents repeatedly play one of the equilibria twice and the other equilibrium once. So agents that use PR are able to converge to patterns of Nash equilibria that have been observed by human subjects in experiments.

My main focus was to develop a learning algorithm that could sustain these patterns of Nash equilibria, which PR is able to do. While these alternations have appeared in the experimental data, there are no experiments on the continuous version of the battle of the sexes game. Also the experiments on the discrete version typically have other aspects such as communication. Therefore, it is difficult to compare the results of these simulations with the current literature to any level beyond the presence of these patterns.

4.2 Experimental Hypotheses

Next, we run simulations using the algorithm on the minimum effort coordination game and develop testable experimental hypotheses. The benefit of using computational agents is that simulations are essentially costless, which allows us to run many trials for each parameter value.

Previous experiments on the minimum effort coordination game have focused on differences in cost and group size. The experiments have typically compared two different parameter values: a low and high cost or a small and large group (Goeree & Holt 2005). Experiments examining a large set of parameters are difficult due to constraints on the number of subjects in a given subject pool, as well as monetary costs for running large experiments. Simulations using the algorithm provide a testbed to simulate these experiments for many different parameter values. Unlike the binary comparisons, examining a larger set of parameters will give us a better understanding of the behavior which may have been overlooked in the past.

From the minimum effort coordination game defined above in equation ??, we run simulations with $\alpha = 1$, $\delta = 0$, $s_i \in [0, 1]$ for groups of four agents with 9 different costs, varying from $c = 0.1$ to $c = 0.9$. At each parameter value, we run 300 simulations lasting for 50 rounds.

Convergence Point: We find that higher costs lead to lower convergence points. Convergence points are the average play over the last 10 periods of the repeated game. The convergence points

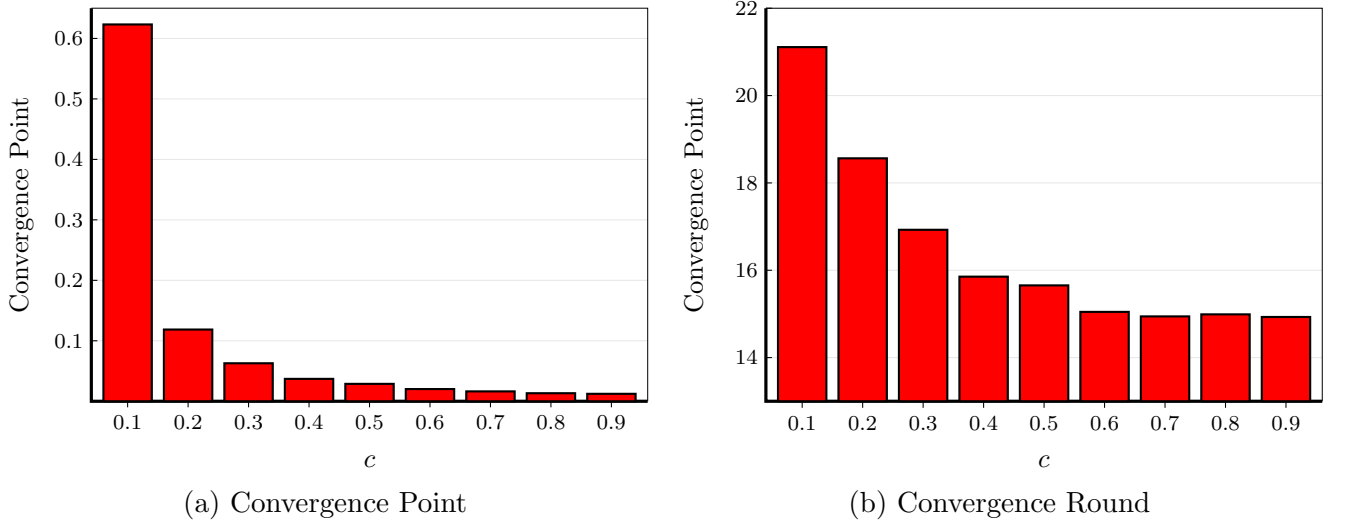


Figure 6: Convergence Points and Convergence Round as a Function of Cost

of these simulations are displayed in Figure 6(a). This is consistent with experimental results from minimum effort coordination games as shown in Goeree & Holt (2005).

Convergence Speed: We then examine the effect of different costs on speed of convergence.⁴ Based on the simulations, we find that the number of rounds required to converge decreases with c , so convergence is faster when the cost is higher. A plot of convergence as a function of c is displayed in Figure 6(b) (higher bars mean slower convergence). The intuition for increase in speed of convergence for higher cost is simple; it is more expensive for agents to search for different outcomes or experiment with different strategies.

Average Payoff: These convergence results have some interesting effects on the agents' payoffs. When agents do not all choose the same effort (i.e., best respond), the outcome is pareto inefficient. If all agents chose the minimum effort for a given strategy profile, then everyone's payoff would be weakly higher, with at least one receiving a strictly higher payoff. Since it is inefficient when all agents are not choosing the same effort, slow convergence may lead to lower average payoffs. The average payoff per agent for different costs is displayed in Figure 7. It is difficult to compare the welfare between two experiments with different costs because they have different payoff functions. Even though welfare is difficult to compare, the payoff for any given strategy profile is lower when the cost of effort is higher. Intuition thus suggests that higher cost of effort should lead to lower average payoffs in the repeated game. However, the simulations suggest that higher cost may actually lead to higher payoffs.

On one hand, with higher costs the agents receive lower payoffs for similar strategy profiles. In addition, the agents are converging to lower effort, which also leads to lower payoffs. However, the simulations show that high costs yield faster convergence, which eliminates some of the inefficiency caused by non-coordination, and increases the agents payoffs. In fact, the increase in payoffs due to faster convergence outweighs the decrease in payoffs due to higher cost and lower convergence

⁴We use convergence in γ as a measure of convergence.

point. Note that the difference in average payoff shrinks as number of rounds increases in Figure 7. This result is due to the fact that the inefficiencies of non-convergence at the beginning of the repeated interaction become less important as the number of rounds increases since it is more likely that the agents have converged.

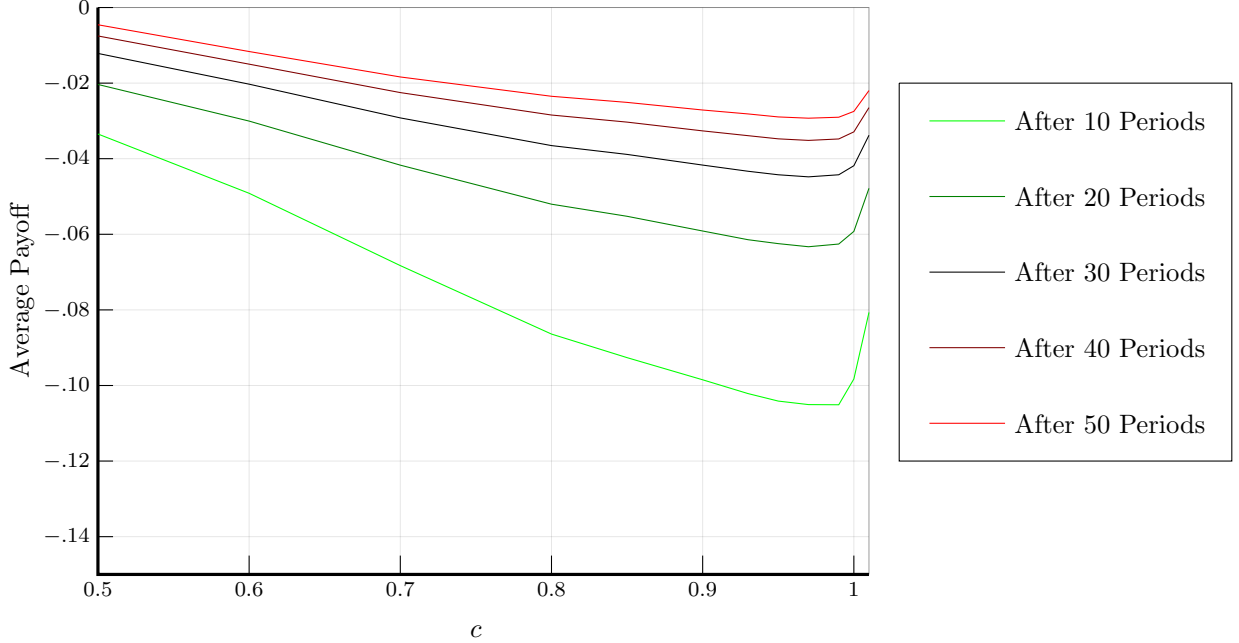


Figure 7: Average Payoffs for Different Costs in Minimum Effort Coordination Game as a Function of Number of Rounds

Based on the simulations, we test the following three hypotheses in the experimental laboratory:

Hypothesis 1. Convergence Point: The game will converge to a pareto dominated payoff as the cost of effort increases.

Hypothesis 2. Convergence Speed: The game will converge faster to an equilibrium as the cost of effort increases.

Hypothesis 3. Average Payoff: The average payoff does not monotonically decrease as the cost of effort increases.

5 Experiments

In this section we describe the setup and results for the experiments that will test the hypotheses formulated from the simulations.

5.1 Design

5.1.1 Overview

The experiments were conducted at the California Social Science Experimental Laboratory (CASSEL) located in the University of California, Los Angeles (UCLA). A total of 60 subjects participated in the experiments. The average performance-based payment was 20USD. All students were registered as subjects with CASSEL (signed a general consent form) and the experiment was approved by the local research ethics committee at both universities. These labs consist of over 30 working computers divided into a cubicles, which prevents students from viewing another student's screen.

The experiment was programmed and conducted with the experiment software z-Tree (Fischbacher 2007). The instructions were available both in print as well as on screen for the participants, and the experimenter explained the instruction in detail out-loud. Participants were also given a brief quiz after instruction to insure proper understanding of the game and the software. A copy of the instruction, as well as the payoff tables, are available in the Appendix.

The subjects were randomly assigned to their roles in the experiment. Furthermore, no one participated in more than one experiment. The identity of the participants as well as their individual decisions were kept as private information. However, each groups knew their own minimum effort. Experiment used fictitious currency called francs. The participants were fully aware of the sequence, payoff structure, and the length of the experiment. All participants filled out a survey immediately after the experiment.

5.1.2 Details of the Experiment

A total of 20 subjects participated in each session. These 20 subjects were split into 5 groups of 4, and each group used a different cost parameter. The entire session was divided into 5 blocks, and each block was divided into 15 rounds. After each block, the subjects were randomly rematched (with replacement) to another group of 4 and were randomly reassigned another payoff parameter (with replacement). See Figure 8 for the time line.

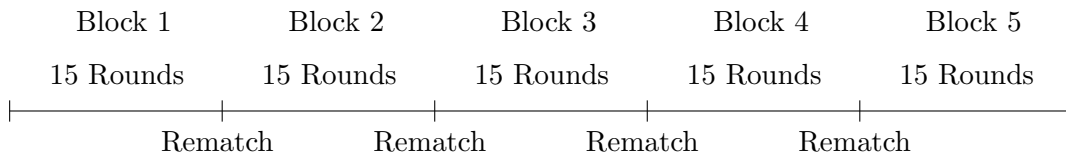


Figure 8: Timeline and Matching Structure for the Experiment

Subjects played a minimum effort coordination game per round. Their task was to choose an effort level,

$$s_i \in \{1, \dots, 7\}$$

and their payments were determined by the following payoff function

$$p_i = 1000 \left(\min_{j \in N} \{s_j\} \right) - c(s_i) + 5950$$

In each block, there were 5 groups each with a different payoff matrix based on

$$c \in \{50, 500, 900, 950, 990\}$$

The subjects were shown the payoff table displayed in Table 1, with the calculation already completed for the subjects. The group size, randomization, and the fact that everyone in the group were using the same payoff table were common knowledge. However, the group's own minimum effort was private information to the group and was not available to the outside members.

	Minimum effort of all agents						
<i>i</i> 's Effort	7	6	5	4	3	2	1
7	$12950 - 7c$	$11950 - 7c$	$10950 - 7c$	$9950 - 7c$	$8950 - 7c$	$7950 - 7c$	$6950 - 7c$
6	—	$11950 - 6c$	$10950 - 6c$	$9950 - 6c$	$8950 - 6c$	$7950 - 6c$	$6950 - 6c$
5	—	—	$10950 - 5c$	$9950 - 5c$	$8950 - 5c$	$7950 - 5c$	$6950 - 5c$
4	—	—	—	$9950 - 4c$	$8950 - 4c$	$7950 - 4c$	$6950 - 4c$
3	—	—	—	—	$8950 - 3c$	$7950 - 3c$	$6950 - 3c$
2	—	—	—	—	—	$7950 - 2c$	$6950 - 2c$
1	—	—	—	—	—	—	$6950 - c$

Table 1: Sample Payoff Table that was used in the Experiment

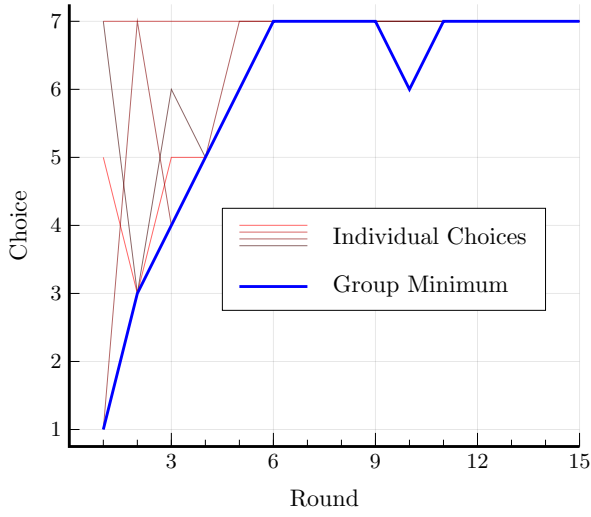
Calculations were already filled in for the subjects

5.2 Results

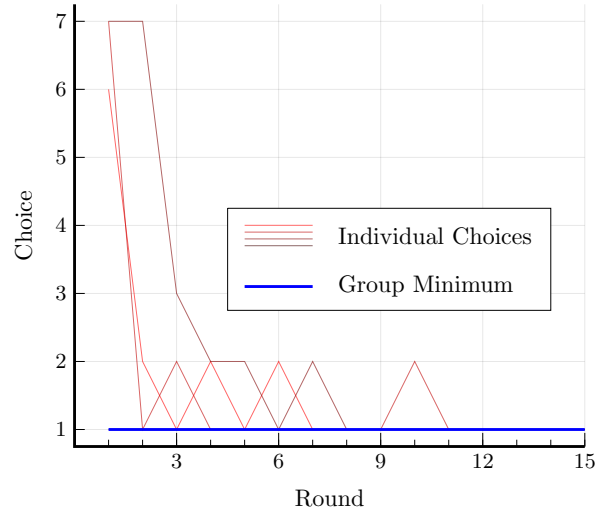
Figure 9 illustrates sample results from one of the block of sessions. Figure 9 (a) is an example where there is a high level of coordination (converging to an effort level of 7) and Figure 9 (b) is an example where there is a low level of coordination (converging to an effort level of 1).

5.2.1 Convergence Points

First, we test the hypothesis that higher costs will lead to lower convergence points and provide the results in Table 2, Table 3, and Figure 10. These results are taken from the average choice of the last 5 rounds and it supports the hypothesis that the average choice drops as the cost parameter increases. While the cost parameter between $c \in \{50, 500\}$ provides a high level of



(a) Coordination to High Effort



(b) Coordination to Low Effort

Figure 9: Sample Results From One of The Block of Session for Illustration Purpose
The thin lines represent individual choices and the thick line represents the group's minimum choice

average choice around 4.5 to 5, the average choice drops significantly lower to 1 to 1.2 for cost parameter between $c \in \{900, 950, 990\}$. Although there is not a significant difference between the means from $c = 900$ and $c = 950$, the differences are significant in the right direction for the rest of the mean comparisons.

	c = 50	c = 500	c = 900	c = 950	c = 990
Choice	4.8485	4.5000	1.2864	1.2606	1.1242
SE	0.0932	0.0975	0.0363	0.0391	0.0244

Table 2: Average Choice for Different Cost Parameters

	$\mu_{50} > \mu_{500}$	$\mu_{500} > \mu_{900}$	$\mu_{900} > \mu_{950}$	$\mu_{950} > \mu_{990}$
p-value	0.0126	0	0.1677	0.0023
t-value	2.2428	21.2006	-0.9642	2.8422

Table 3: Average Choice Comparison

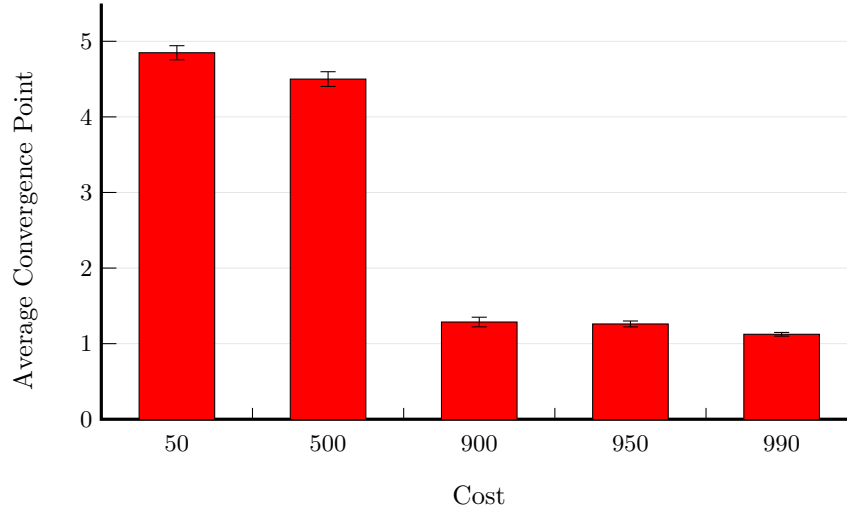


Figure 10: Average Convergence Points for Different Cost Parameters

5.3 Convergence Speed

Comparing convergence speed is bit trickier than comparing convergence points. In the simulations, the agents are computer programs, so it is easy to see why they are making certain choices. However, with humans it is not as straight forward. Consider the following example in Figure 11. If one were to use a rule that the convergence occurs when there are no deviations (i.e., everyone is best responding), then there won't be any convergence until round 13 in the example. When studying experimental results with subjects from a laboratory, this may be too conservative of a criterion. Noisy choice in human behavior is often expected in experiments. Whether these noises are rational or not is another story. However, there are many different ways of modeling noisy choices, such as the Quantal Response Equilibrium (McKelvey & Palfrey 1995), the Level-K Model, and the Cognitive Hierarchy Model (Camerer, Ho & Chong 2004), among others.

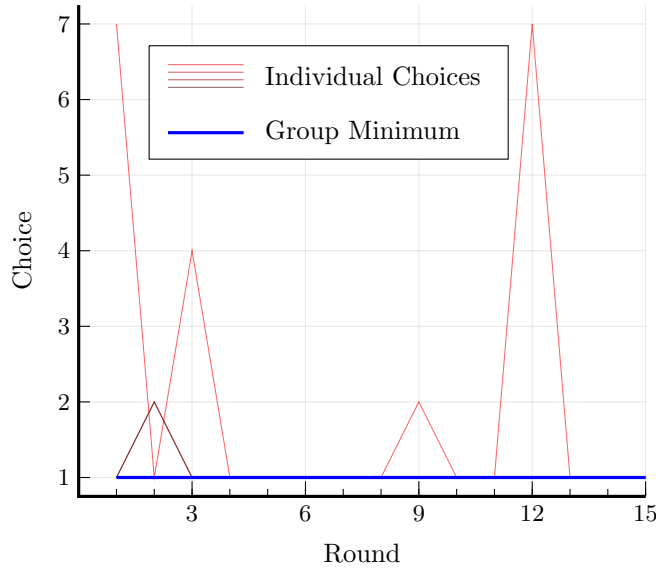


Figure 11: A Sample Result From a Block of Session

The thin lines represents individual choices and the tick line represents the group's minimum choice

Here, we provide two means of measuring convergence. First, we use a more quantitative measure of convergence called v-bounded condition. Then we introduce a more qualitative and intuitive measure of convergence called the similarness condition.

Definition: The game has converged to a particular equilibrium at round t under *v-bounded condition* if the variance of efforts chosen is always less than v for every round starting from t . Specifically, $var_{t+m}(\sigma_1, \dots, \sigma_n) \leq v, \forall m \geq 0$.

For example, if the strategy profile σ consists of $[3, 3, 3, 4]$, this will require that a variance parameter of $v \geq 0.25$ will be needed to consider this strategy profile as converged under the v-bounded condition. See Table 4 for other samples of strategy profile and its required variance parameter for v-bounded condition.

Using the v-bounded condition criterion for the notion of convergence, Figure 12 illustrates the average rounds it took for the game to converge.⁵ Although convergence speed seems to be

⁵We drop the last round deviation because there may be end game effects.

σ	Minimum v
[3, 3, 3, 4]	.25
[3, 3, 4, 4]	.33
[2, 3, 3, 4]	.66
[3, 3, 4, 4]	.92
[3, 3, 3, 5]	1

Table 4: Samples of Strategy Profile and its Required v Parameter for v -bounded Condition

increasing as the cost parameter increases, differences are not statistically significant. Consider the following example from Figure 11 to illustrate why the v -bounded condition may not be a good criterion: the condition requires $v \geq 9$ in order to allow this particular example to be considered converged due to a large jump in choice of effort by one of the players in round 12. This does not take into account that the deviation is by one person for only one period. However, intuitively, one may think that this game has converged at round 4.

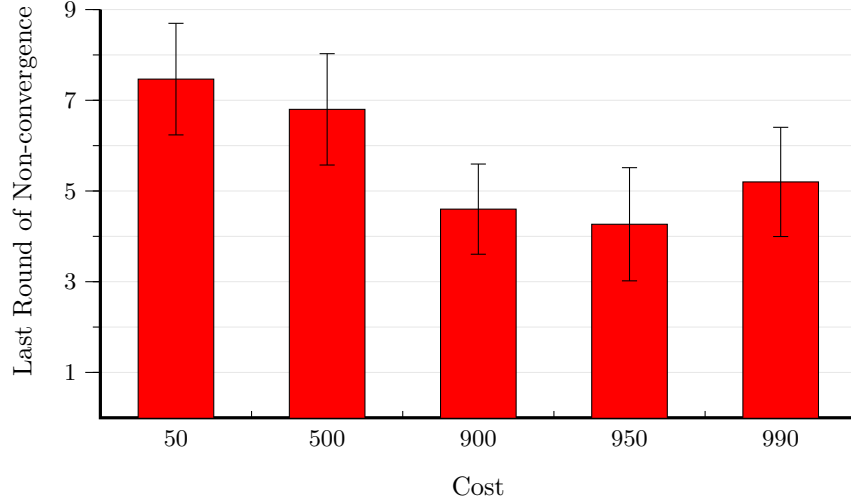


Figure 12: Number of Rounds Needed for Convergence for $v \leq 0.5$

Therefore, we use a more intuitive and qualitative measure of convergence. For a given round, look at the number of different effort in the strategy profile. Then, the game has converged to a particular equilibrium if a high proportion of people use the same strategy. We call this the *similarity condition*. The added benefit of the similarity condition is that it does not unreasonably penalize cases where one person may deviate significantly away from the best response for just one period. By the same token, it also means that this measure treats the following two strategy profiles as equally converged: [2, 2, 2, 3] and [1, 1, 1, 7].

Figure 13 shows the frequency of different strategies played for various cost of effort. If the game is indeed converging faster under the similarity condition, we expect to see a higher frequency of

blue and sky-blue, which indicates everyone playing the same strategy and three people playing the same strategy, respectively. As the cost of effort increases, we observe an increase in frequency of blue and sky-blue. This increase in frequency holds true for any given round. Furthermore, the frequency of blue and sky-blue also increases as the experiment proceeds (number of round increases). In other words, there are many different strategies being played in the initial round but subjects learn to best respond.

This similarness condition as a convergence criterion provides support that the game converges faster to a particular equilibrium as the cost of effort increases.

5.4 Average Payoff

Finally, we analyze the behavior of the average payoff as the cost increases. Refer to Figure 14 and Table 5 and 6 to see the average payoff and their mean comparisons up to 4 rounds for each of the cost parameters from the experiment. We find a statistically significant decrease in average payoff from $\mu_{50} = 9088$ at $c = 50$ to $\mu_{950} = 4846$ at $c = 950$ ($p \approx 0$). However, as the simulation has predicted, the average payoff at $c = 990$ of $\mu_{990} = 5136$ is significantly higher than the average payoff at $c = 950$ of $\mu_{950} = 4846$ ($p < 0.05$). This suggests that early in the interaction the average payoffs are not monotonically decreasing as the costs increases

Given that we observe a non-monotonicity in average profit as a function of cost of effort in the first 4 rounds, we test the significance after the entire block of the experiment (15 rounds). The result is displayed in Figure 15 and Table 7 and 8. Again, we observe a similar pattern to the results from the first 4 rounds. The average payoff of $\mu_{990} = 5650$ at $c = 990$ is significantly greater than the average payoff of $\mu_{950} = 5560$ at $c = 950$ ($p < 0.1$). Furthermore, the average payoff in this setting is the lowest at $c = 950$, which is also lower than the average payoff of $\mu_{900} = 5652$ at $c = 900$ ($p < 0.1$).

While the p-values for the non-monotonicity hypothesis are weaker after 15 rounds than after 4 rounds, this is what the simulations predicted. The difference between the average payoff when $c = 990$ and $c = 950$ diminishes as more rounds are played. This confirms the prediction made by the simulation in Figure 7. As more rounds are played, the positive welfare from the lower cost averages out the negative welfare from the wasted effort. For example, after 4 rounds, the difference in average payoff is $\mu_{990} - \mu_{950} = 288.9583$. But, after 15 rounds, the difference decreases to $\mu_{990} - \mu_{950} = 90.1889$. In other words, the non-monotonicity of average payoff is most salient at the initial phase of the game.

6 Conclusion

We have developed a learning algorithm which can be implemented with computational agents. We have also shown that simulations with these computational agents are able to generate data which shares many features with experimental data for both the minimum effort coordination game and the battle of the sexes game. In particular, agents using my algorithm are able to sustain alternations between Nash equilibria in the battle of the sexes game: a result which previous learning models have had a difficult time sustaining. We have used these agents to run simulations for a wide class of parameters in both of these games, and developed testable hypotheses about human behavior in these games. We have designed an experiment based on these simulations and confirmed the testable hypotheses developed from the simulations.

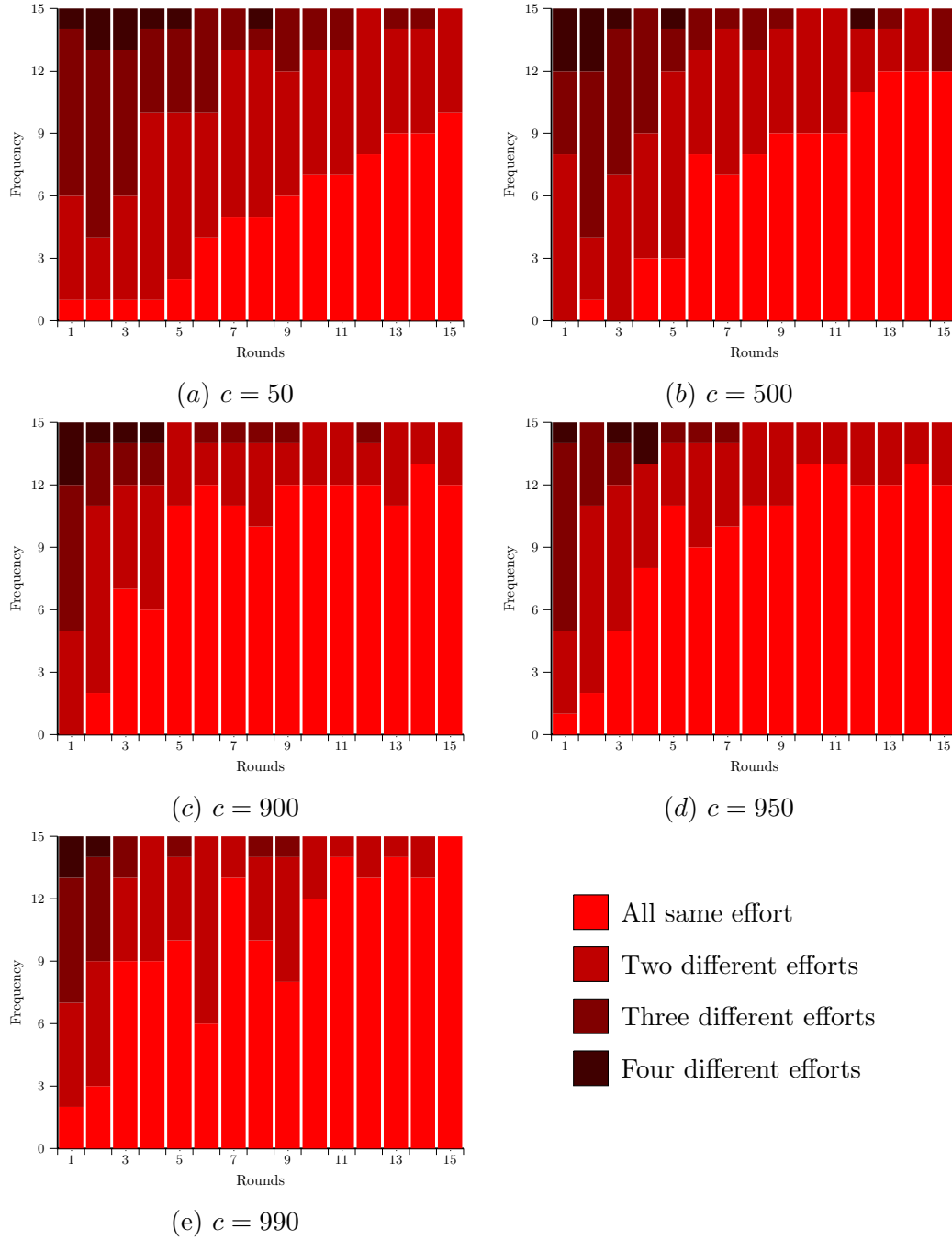


Figure 13: Frequency of Different Strategies Played for Various Costs

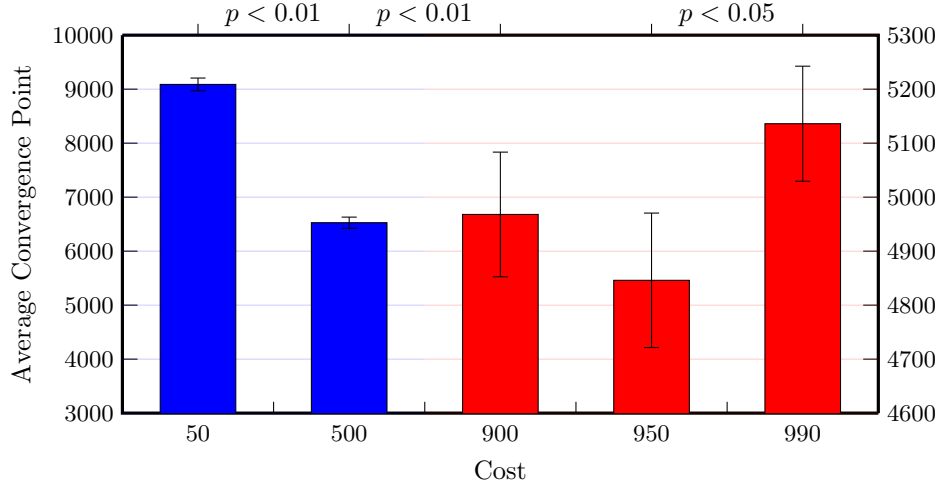


Figure 14: Average Payoff After 4 Rounds

	c = 50	c = 500	c = 900	c = 950	c = 990
μ	9088	6527	4968	4846	5136
SE_{μ}	118.05	103.36	115.42	124.51	106.56

Table 5: Average Payoffs for Different Cost Parameters After 4 Rounds

	$\mu_{50} > \mu_{500}$	$\mu_{500} > \mu_{900}$	$\mu_{900} > \mu_{950}$	$\mu_{950} < \mu_{990}$
p-value	0	0	0.2366	0.0393
t-value	16.3234	10.05	0.7178	1.7632

Table 6: Average Payoffs Comparison After 4 Rounds

	c = 50	c = 500	c = 900	c = 950	c = 990
μ	9791	7489	5652	5560	5650
SE_{μ}	76.23	53.75	38.61	43.66	35.02

Table 7: Average Payoffs for Different Cost Parameters After 15 Rounds

	$\mu_{50} > \mu_{500}$	$\mu_{500} > \mu_{900}$	$\mu_{900} > \mu_{950}$	$\mu_{950} < \mu_{990}$
p-value	0	0	0.0557	0.0536
t-value	24.6797	27.75	1.5928	1.6114

Table 8: Average Payoffs Comparison After 15 Rounds

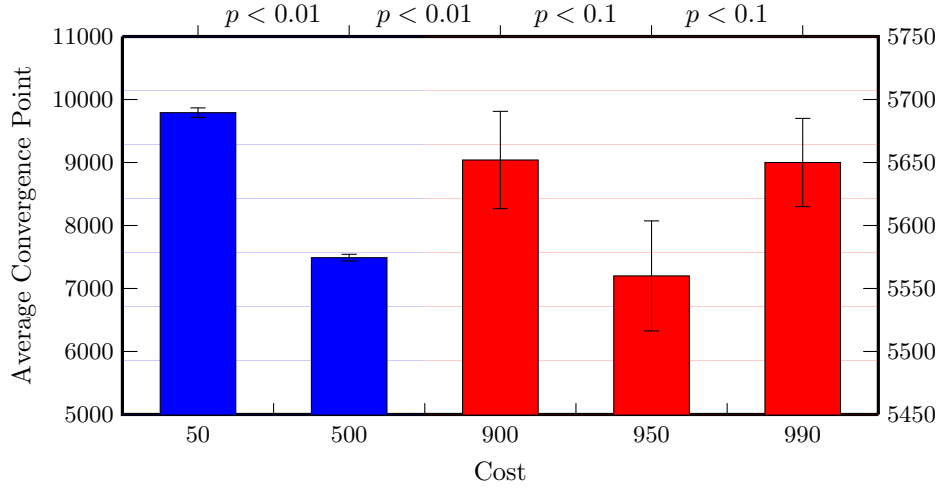


Figure 15: Average Payoff After 15 Rounds

The validity of my algorithm has been tested with experimental evidence from two coordination games. Ideally, possibly with modifications, this algorithm will provide results consistent with a much wider class of games. The wider the class of games that this type of algorithm is consistent with, the more reliable the results will be for simulations of new games. As these agents become a more reliable predictor of human behavior, they can become very valuable asset to experimental economists. In the beginning we can replace costly pilot experiments by running simulations of experiments with these agents to determine optimal parameter values for experiments. As these agents become more reliable, we may be able to use these agents to interact with human subjects during experiments, providing larger subject pools. Eventually, entire experiments could be run with these computational agents, and when a useful result is found, it can be confirmed experimentally with human subjects.

References

- Arifovic, J. & J. Ledyard. 2005. "Computer testbeds for mechanism design." *manuscript, April*.
- Berger, Ulrich. 2007. "Two more classes of games with the continuous-time fictitious play property." *Games and Economic Behavior* 60(2):247 – 261.
*<http://www.sciencedirect.com/science/article/B6WFW-4MV1P0N-1/2/4546c5528b91f2ea2747bd6b18976788>
- Brown, G.W. 1951. "Iterative solution of games by fictitious play." *Activity analysis of production and allocation* 13:374–376.
- Camerer, Colin F., Teck-Hua Ho & Juin-Kuan Chong. 2004. "A Cognitive Hierarchy Model of Games*." *Quarterly Journal of Economics* 119(3):861–898.
*<http://www.mitpressjournals.org/doi/abs/10.1162/0033553041502225>
- Fischbacher, Urs. 2007. "z-Tree: Zurich toolbox for ready-made economic experiments." *Experimental Economics* 10(2):171–178.
*<http://dx.doi.org/10.1007/s10683-006-9159-4>
- Friedman, Daniel. 1991. "Evolutionary Games in Economics." *Econometrica* 59(3):637–666.
*<http://www.jstor.org/stable/2938222>
- Goeree, Jacob K. & Charles A. Holt. 2005. "An experimental study of costly coordination." *Games and Economic Behavior* 51(2):349 – 364. Special Issue in Honor of Richard D. McKelvey.
*<http://www.sciencedirect.com/science/article/B6WFW-4DR1NP7-4/2/3dc5377d897b9ef44cd71a70f4c00ca4>
- Hahn, Sunku. 2008. "The convergence of fictitious play in games with strategic complementarities." *Economics Letters* 99(2):304 – 306.
*<http://www.sciencedirect.com/science/article/B6V84-4PB6VRM-4/2/78d427466e289cf2116e63d0cdc86a2e>
- Hart, Sergiu. 2002. "Evolutionary dynamics and backward induction." *Games and Economic Behavior* 41(2):227 – 264.
*<http://www.sciencedirect.com/science/article/B6WFW-4742689-1/2/889fcfb46755ea4e2923684a544dc17b>
- Kandori, Michihiro, George J. Mailath & Rafael Rob. 1993. "Learning, Mutation, and Long Run Equilibria in Games." *Econometrica* 61(1):29–56.
*<http://www.jstor.org/stable/2951777>
- Knez, Marc & Colin Camerer. 1994. "Creating Expectational Assets in the Laboratory: Coordination in 'Weakest- Link' Games." *Strategic Management Journal* 15:101–119.
*<http://www.jstor.org/stable/2486813>
- Lambson, Val E. & Daniel A. Probst. 2004. "Learning by matching patterns." *Games and Economic Behavior* 46(2):398 – 409.
*<http://www.sciencedirect.com/science/article/B6WFW-48XD3T9-4/2/20049486962c2cf3348a7ee0518c0cc1>

- McKelvey, R.D. & T.R. Palfrey. 2001. Playing in the dark: Information, learning and coordination in repeated games. Technical report Technical report, California Institute of Technology, 2001.
- McKelvey, Richard D. & Thomas R. Palfrey. 1995. "Quantal Response Equilibria for Normal Form Games." *Games and Economic Behavior* 10(1):6 – 38.
[*http://www.sciencedirect.com/science/article/B6WFW-45NJFPR-10/2/4d59f8a5a50e7a4eaf62f539c9c66cfd](http://www.sciencedirect.com/science/article/B6WFW-45NJFPR-10/2/4d59f8a5a50e7a4eaf62f539c9c66cfd)
- Milgrom, Paul & John Roberts. 1991. "Adaptive and sophisticated learning in normal form games." *Games and Economic Behavior* 3(1):82 – 100.
[*http://www.sciencedirect.com/science/article/B6WFW-4CYGSTV-32/2/f4f5ed6e8354c1f82a3c869b3fdbd131](http://www.sciencedirect.com/science/article/B6WFW-4CYGSTV-32/2/f4f5ed6e8354c1f82a3c869b3fdbd131)
- Miyasawa, K. 1961. "On the convergence of the learning process in a 2 x 2 non-zero-sum two-person game." *Economic Research Program, Research Memorandum* 33.
- Monderer, Dov & Lloyd S. Shapley. 1996. "Fictitious Play Property for Games with Identical Interests." *Journal of Economic Theory* 68(1):258 – 265.
[*http://www.sciencedirect.com/science/article/B6WJ3-45NJJNP-29/2/0909c553a060e957fecf397a60a2b9c0](http://www.sciencedirect.com/science/article/B6WJ3-45NJJNP-29/2/0909c553a060e957fecf397a60a2b9c0)
- Rapoport, A., M. Guyer & D.G. Gordon. 1976. *The 2 x 2 game*. Univ of Michigan Pr.
- Smith, J. Maynard & G. R. Price. 1973. "The Logic of Animal Conflict." *Nature* 246(5427):15–18.
[*http://dx.doi.org/10.1038/246015a0](http://dx.doi.org/10.1038/246015a0)
- Sonsino, Doron. 1997. "Learning to Learn, Pattern Recognition, and Nash Equilibrium." *Games and Economic Behavior* 18(2):286 – 331.
[*http://www.sciencedirect.com/science/article/B6WFW-45M8X2C-1G/2/e45ed56ef1a63d31d15f6738acafe776](http://www.sciencedirect.com/science/article/B6WFW-45M8X2C-1G/2/e45ed56ef1a63d31d15f6738acafe776)
- Sonsino, Doron & Julia Sirota. 2003. "Strategic pattern recognition—experimental evidence." *Games and Economic Behavior* 44(2):390 – 411.
[*http://www.sciencedirect.com/science/article/B6WFW-48M7V5K-J/2/d3cc291c66e7caa0ab1557de7a6ca2f6](http://www.sciencedirect.com/science/article/B6WFW-48M7V5K-J/2/d3cc291c66e7caa0ab1557de7a6ca2f6)
- Van Huyck, John B., Raymond C. Battalio & Richard O. Beil. 1990. "Tacit Coordination Games, Strategic Uncertainty, and Coordination Failure." *The American Economic Review* 80(1):234–248.
[*http://www.jstor.org/stable/2006745](http://www.jstor.org/stable/2006745)
- Young, H. Peyton. 1993. "The Evolution of Conventions." *Econometrica* 61(1):57–84.
[*http://www.jstor.org/stable/2951778](http://www.jstor.org/stable/2951778)

Appendix

A Simulation Parameters

A.1 FP Simulations

The initial choice in the fictitious play simulations is chosen from $U[0, 1]$. After this players best respond to empirical distribution of the history of play. In the continuous minimum effort coordination game, the best response to any history of choices will be either 0, 1 or one of the previous choices. To calculate this best response, the players cycle through all possibilities and keep the one that gives the highest payoff.

A.2 AL Simulations

In the simulations from Arifovic & Ledyard’s (2005) algorithm, each agent keeps a collection of choices. The size of the collection is $J = 50$. The collection is initialized by drawing J numbers from $U[0, 1]$. At each round, the agent’s action is chosen randomly from the collection based on a probability distribution weighted by the forgone utility⁶ of each action. The agents then update their choice set by experimentation and replication of the good strategies. The experimentation parameter is $\rho = 0.03$. After the choice set is updated, the probability distribution is updated for the following round, and then the agent is ready to make his choice for the next round. A further explanation of the AL algorithm is available in their paper.

A.3 PR Simulations

The PR simulations are described in detail in the body of the paper. All of the simulations used the following parameters. Agents take $J = 10$ random samples when updating their quasi-best-response and their highest and lowest known payoffs. The boundaries for the confidence parameter are $\gamma \in [10, 10000]$. The parameter relating variance and confidence is set to $\rho = 12$. When updating the confidence, the agents have $\alpha = 1/3$ and $\alpha_2 = 1/6$. This means that the agents’ confidence increases when their prediction is within $1/6$ of the actual outcome. When the agents are recognizing patterns, they look for the $j = 3$ most similar plays from the past to make their prediction. Finally, the agents won’t start a moving session unless their confidence parameter is at least $\gamma_{MS} = 50$. Once the agents are in a moving session, they are allowed $\bar{f} = 4$ failures before they become discouraged and stop the moving session.

⁶What the payoff would have been if the agent had chosen this instead of what they chose last round.

B Experimental Instructions

Procedural Summary

There are five papers folded in half on each of your desk. They are labeled “Table 1”, “Table 2”, “Table 3”, “Table 4” and “Table 5”. Each of these papers will present a different payoff table. The payoff from your paper labeled “Table 1” will be identical to everyone else’s paper labeled “Table 1”, the payoff from your “Table 2” will be identical to everyone else’s “Table 2” and so on. Please do not open any of these papers until you are instructed to do so.

The experiment consists of 5 blocks and each block will consist of 15 periods. You will be placed in a group with 3 others who are randomly chosen for each block. This means that at the beginning of each block, you will be randomly matched with 3 others individuals to be placed in a group size of 4.

Your payoff from each period will be determined by one of the 5 folded papers. You will be using one payoff table for the entire block. Everyone in your group will be using the exact same payoff table as you. Your group members are randomly chosen (via random number generator) and your payoff tables for each block are also randomly chosen (via random number generator). Your payment will be sum of your entire earnings from all 5 blocks.

Q: How do you know what period you are in? What Payoff table to use? Which block you are in?

A: Refer to Figure 1. Number of periods is denoted in the upper left corner. The first line of the computer’s instruction tells you what table your payoff is determined from.

Periods 1-15 is block one. Periods 16-30 is block two. Periods 31-45 is block three. Periods 46-60 is block four. Periods 61-75 is block five.

Note: There will be no sign that tells you that you are starting a new block. So pay attention to the period numbers.

Q: Is it possible to have the same table number from one block to another?

A: Yes, this is because you are randomly assigned a payoff table for each block.

Timeline and Summary

Block 1 begins. You’re randomly matched with 3 other people and everyone in your group is randomly assigned a same payoff table to use for this block. You may open the payoff table at this time. The period will begin and all the participants will make their choices privately through the computer. Press “okay” after you have made your decision. After all the participants have inputted their choices, the lowest number chosen from your group as well as your payoff will be displayed. Press continue to move on to the next period. You and your group will do this for 15 periods and block 1 will come to an end.

Block 2 begins. You’re randomly matched with 3 other people and everyone in your group is randomly assigned a same payoff table to use for this block. You may open the payoff table at this time. The period will begin and all the participants will make their choices privately through the computer. Press “okay” after you have made your decision. After all the participants have inputted their choices, the lowest number chosen from your group as well as your payoff will be displayed. Press continue to move on to the next period. You and your group will do this for 15 periods and block 2 will come to an end.

This continues until all 5 blocks are played. Your payment will be the sum of your payoff from each period in all 5 blocks.

Experiment Overview

You are about to participate in an experiment in the economics of decision making. If you listen carefully and make good decisions, you could earn a considerable amount of money that will be paid to you in cash at the end of the experiment.

Please do not talk or communicate with other participants. Feel free to ask questions by raising your hand or signaling to the experimenter.

You will be working with a fictitious currency called Francs. The exchange rate will be specified in the instructions. You will be paid in cash at the end of the experiment.

The experiment consists of a sequence of periods and blocks. There will be total of 5 blocks. For each block, there will be total of 15 periods.

Specific Instructions for Each Period

Exchange rate: _____ Francs = _____ USD.

Your group will consist of you and 3 other individuals (total of 4 people in your group). Your job is to choose one of the following numbers: {1, 2, 3, 4, 5, 6, 7}. The number you choose will remain anonymous. Your individual payoff is determined by your choice and the choice of others in your group. The following is a sample payoff table for illustration purposes only. Your actual payoff table will be using different numbers from this table. The overall ideal will be the same, however.

Table 1: Your payoff in francs

Your choice of number	Lowest choice of number from your group (including you)						
	7	6	5	4	3	2	1
7	25	21	17	13	9	5	1
6	-	23	19	15	11	7	3
5	-	-	21	17	13	9	5
4	-	-	-	19	15	11	7
3	-	-	-	-	17	13	9
2	-	-	-	-	-	15	11
1	-	-	-	-	-	-	13

Examples

- You chose 5 and the lowest choice of number from your group is 5. Then you win 21 francs.
- You chose 4 and the lowest choice of number from your group is 2. Then you win 11 francs.
- You chose 3 and the lowest choice of number from your group is 2. Then you win 13 francs.

Quiz

You chose 2 and the lowest choice of number from all the participants is 1. Then you win _____ francs.

Any questions?

The screenshot shows a game interface with a light gray background. At the top, there is a header bar with the word "Period" on the left and "1 out of 50" in the center. Below the header, the main area contains the text "Your payoff is determined by Table 1." followed by "Pick your number: 1, 2, 3, 4, 5, 6, 7" and a small blue rectangular input box. In the bottom right corner, there is a red rectangular button labeled "OK".

Figure 1: Sample Screenshot