Does Playing Against An Error Prone Opponent Influence Learning in Nim? *

C. Nicholas McKinney, Jr.† *and John Van Huyck*‡

Abstract

When learning to play a game well, does it help to play against an opponent who makes the same sort of mistakes one tends to make or is it better to play against a procedurally rational algorithm, which never makes mistakes? This paper investigates subject performance in the game of Nim. We find evidence that subject performance improves more when playing against a human opponent than against a procedurally rational algorithm. We also find that subjects learn to recognize certain heuristics that improve their overall performance in more complex games.

JEL Classification: C72, C92, D83

Keywords: Bounded rationality, learning, heuristics, perfect information, Nim, human behavior, experiment

1 Introduction

When learning to play a game well, does it help to play against an opponent who makes the same sort of mistakes one tends to make or is it better to play against a procedurally rational algorithm, which never makes mistakes. A procedurally rational algorithm is unforgiving. If one makes a mistake, one never gets another chance to win. An error prone opponent may miss one's mistakes and give one multiple chances to find a way to win. This paper reports an experiment to address two questions: (i) Do people learn to recognize winning positions over time? and, (ii) If they do, does playing against human opponents rather than a procedurally rational algorithm influence learning?

The experiment uses an AXA' design. Learning is investigated by comparing subject performance in the first A treatment with the last A' treatment. In the A treatments, subjects play against the procedurally rational algorithm. In the X treatment, subjects either played against a human or against the procedurally rational algorithm. Performance can then be compared across subjects in the final A' treatment. The research hypotheses are that performance in the first and last A treatment will be the same and that playing against humans will not change performance in the last A treatment. Both hypotheses will be formally rejected.

We find evidence that subject performance improves over time. As in McKinney and Van Huyck [2013], the increase in performance is greater in games that can be solved using heuristics that lessen the subjects' reliance on backwards induction. We still find little evidence that subjects completely master even basic heuristics, but we are able to model their performance in a way that shows that

*Chris Ball provided able research assistance. Patrick Williams provided aid and advice on software development. Jana Joswiak and Derek Le Blanc helped with recruiting and accounting. The Private Enterprise Research Center and the National Science Foundation provided financial support. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Private Enterprise Research Center or the National Science Foundation. First draft: February 2001; This draft: March 2014.

†Department of Economics, Rhodes College, Memphis, TN 38112

‡Department of Economics, Texas A&M University, College Station, TX 77843

subjects that never play against other humans do not perform as well in the last set of games as those that do. Playing against an error prone opponent does improve learning.

2 Analytical Framework

To focus the analysis, consider the game of Nim. Nim is a two player game in which the players alternate taking one or more stones from one of m rows. The player who takes the last stone wins. A game of Nim can be summarized by a $1 \times m$ vector of natural numbers, g, with elements g_i denoting the number of stones in row i. Let G denote the set of all Nim games, where $g = \{g \in \aleph^m | m \in \aleph\}$and $\aleph$ denotes the natural numbers. The rank of Nim game g, $r(g)$, is equal to the number of stones used in the game:

$$r(g) = \sum_{i=1}^{m} g_i$$

Every Nim game $g \in G$ has a game theoretic value that is either a win for the first mover, W, or a loss, L. Let the value function be denoted $v(g) : G \to \{W, L\}$. In addition, procedurally rational algorithms exist for all Nim Games.[1]

Using a procedurally rational algorithm, all Nim games can be categorized into one of two sets: *balanced games*, B, or *unbalanced games,* U. Following McKinney and Van Huyck [2007], convert the decimal representation of the natural numbers in g into the equivalent binary representation. Let $b(g)$ denote the binary representation. Let $d_j(g_i)$ denote the digit in the 2^{j-1} position of $b(g_i)$, where $d_j(g_i) \in \{0, 1\}$. The set of balanced Nim games, B, is

$$B = \left\{ g \in G | \sum_{i=1}^{m} d_j(g_i) \text{ } is \text{ } even \text{ } \forall j \right\}$$

and the set of unbalanced Nim games, U, is

$$U = \{g \in G | g \notin B\}.$$

The value function is then expressed as follows:

$$v(g) = \begin{cases} W & if \quad g \in U \\ L & if \quad g \in B \end{cases}$$

Substantively rational players will win all unbalanced Nim games and lose all balanced Nim games.

Our previous studies have investigated four measures of the complexity of a Nim game.[2] The most intuitive is rank. Nim Games with longer play paths are more difficult to think through than those with shorter play paths. A second measure of complexity is the number of rows or shortest play path through the extensive form. A third measure, NT-complexity, is the number of non-trivial decision nodes in the extensive form of the game. The fourth measure of complexity is the probability a blunderer wins the game playing against a procedurally rational opponent, where a blunderer is a player who chooses a uniformly random feasible action at every information set they are assigned.

McKinney and Van Huyck [2013] found that subject performance was influenced by the presence of heuristics that allow them to play significantly better than a blunderer. The most effective heuristic is move copying (MC): If a player can leave a subgame with two equal rows of stones then she can always win by copying her opponents move in each subsequent subgame until she finally removes the final stone and wins the game. Move copying can easily be expanded to all games with

[1] See Bouton [1901-1902] for the details of the solution algorithm. See Conway [2001] for an alternative solution algorithm.

[2] See McKinney and Van Huyck [2006, 2007] for a more extensive discussion of complexity measures for Nim.

any number of pairs of equal rows. For example, removing one stone from row 3 in game (5,3,4,5) leaves a game with two pairs of equal rows. A player that understands the move copying heuristic should be able to win game (5,3,4,5) against a procedurally rational opponent even if she lacks the ability to analyze the games extensive form and solve it via backwards induction.[3]

3 Experimental Design

The experiment consisted of 4 sessions with 20 participants in each session. Each session contained three treatments. In the first and last Treatments (A and A'), the subjects played against a procedurally rational algorithm. In the middle treatment the subjects either played against one another, treatment B, or against the procedurally rational algorithm, treatment C. Two sessions used the A treatment, B treatment, A' treatment design and shall be denoted ABA' sessions below. Two sessions use the A treatment, C treatment, A' treatment design and shall be denoted ACA' sessions below. Comparing behavior within and between the ABA' and ACA' sessions test for differences in learning.

One of the crucial design issues was controlling the amount of time available to the participants when playing against other subjects or against a procedurally rational algorithm during the middle treatment of a session. The procedurally rational algorithm can be executed almost instantly on the computers available in the Economic Research Laboratory (ERL). The participants had 60 seconds to make their decisions for each game. The subjects' timer only moves when it is their turn to make a decision. If time expires the game is recorded as a loss, the subject earned \$0.10, and the timer restarts for the next game. In the first and last treatments the procedurally rational algorithm took one second to play and thus never timed out. In the middle treatment of the ACA' sessions, the computer program samples from the set of decision times observed in the middle treatment of the ABA' sessions, when our subjects played against each other. As a result, it was possible for the computer implementing the procedurally rational algorithm to time out and lose a winning position in the middle treatment of the ACA' sessions.[4]

In all sessions Treatments A and A' contained 31 games; the 27 games from Treatment 1 of McKinney and Van Huyck [2007] as well as four additional unbalanced games. Of these 31 games, 22 were unbalanced and 9 were balanced. Across the first and last treatments the games differed only in the order of their rows and their order of presentation on the record grid. Subjects were always the first mover. The games ranged in complexity from rank 3 to 17 and had shortest play paths ranging from 1 to 5. Subjects earned \$0.60 for a win and \$0.10 for a loss. Appendix A list the 31 games ordered by NT-complexity.

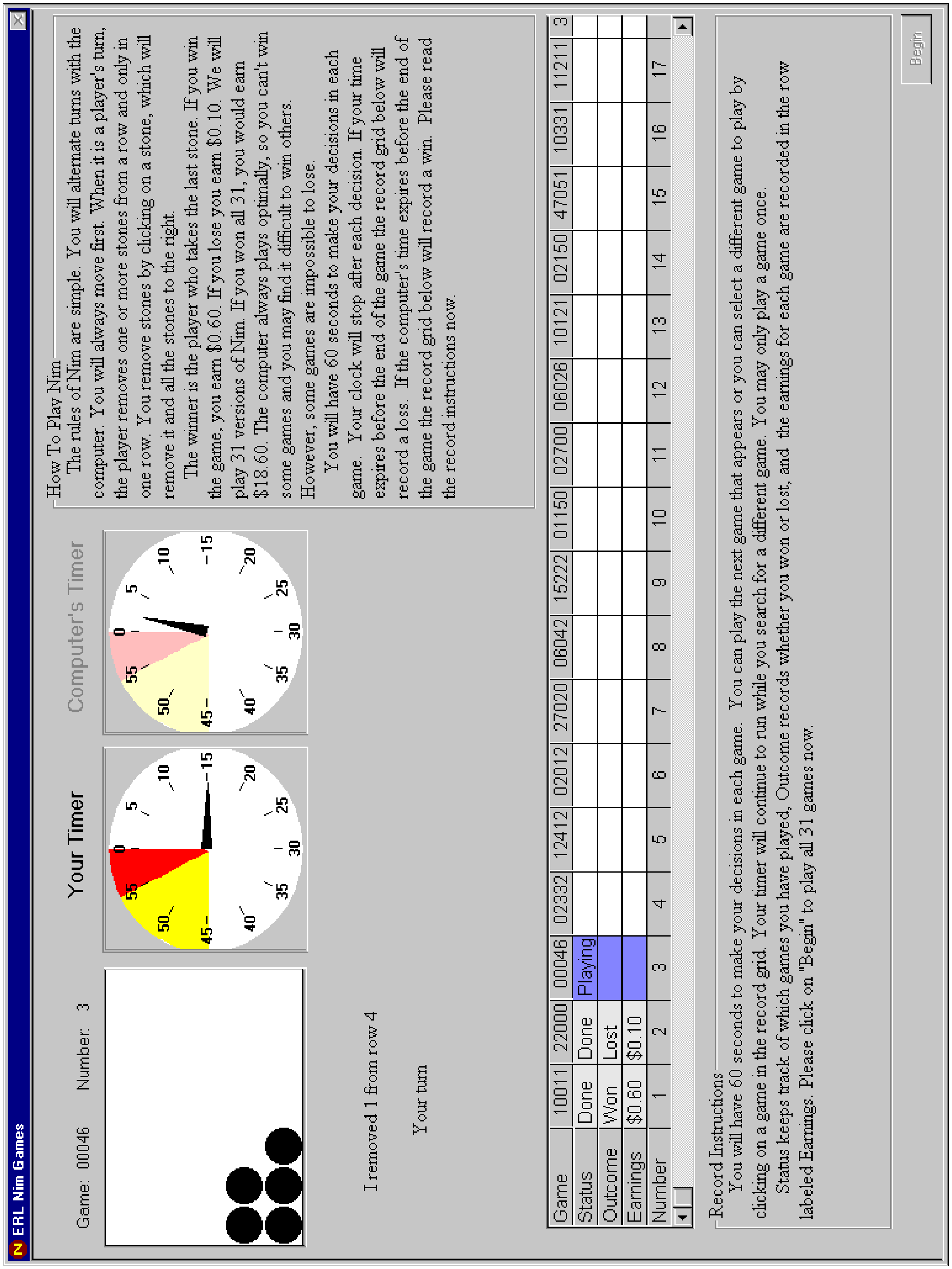
Figure 1 is a screen grab of the graphical interface used in Treatments A and A'. The Nim game was displayed in the upper left corner. Subjects played the game by clicking on a stone (black circle) to remove it and every stone to the right. The timers were displayed to the right of the game grid. Instructions were found to the right of the timers. A dialog reminding the subjects of the current mover and what just happened appears below the game grid. All 31 games were displayed on the record grid at the bottom of the screen, and subjects could choose to play the games in any order. Subjects played at their own pace.

Treatment B (the second session in the ABA' design) was designed to replicate Treatment 1 of McKinney and Van Huyck [2006]. Each subject played 19 pairs of games against each of the other participants in the session. Each pair (match) contained one balanced and one unbalanced game. The 38 games had ranks ranging from 12 to 17. The games were still worth \$0.60 for a win and \$0.10 for a loss. The subjects were paired through a round robin algorithm. At the beginning of each match, subjects were assigned either the role of first mover or second mover. This assignment was based on each subjects' prior assignments. The role of first or second mover did not change

[3] When given the opportunity to play the same game repeatedly Dufwenberg et al. [2010] and Gneezy et al. [2010] show that subjects do learn to backwards induct in games similar to Nim.

[4] There are a total of 14 forfeits by humans in treatment B. The program produced 9 forfeits in treatment C. The humans forfeit 25 times in treatment C.

Fig. 1: Treatment A and A' Graphical User Interface



during a match. The subject who had been assigned the role of first mover fewer times was assigned the role of first mover at the beginning of a new match. In the event of a tie, the role of first mover was assigned at random. The C treatment of the ACA' sessions contained the same games as the B treatment of the ABA' sessions.

Figure 2 is a screen grab of the interface used in Treatment B. The computer's timer is replaced with the other participant's timer. If either timer expires, that participant received \$0.10 for the loss and the other participant received \$0.60 for the win. The record grid at the bottom displays only the games for the current match and all games played in previous matches. Subjects could not move on to the next match until all 20 subjects had completed the current match.

Treatment C (the second treatment in the ACA' session) was identical to Treatment B with one exception; The subjects were not paired against one another. The 38 games were once again played against the procedurally rational algorithm. In order to simulate the humans' deliberation process, the computer now paused before making its decision. In order to best simulate the amount of time that an actual human player spends deliberating each move, the length of the pause is determined by uniformly sampling from the previously recorded times from the ABA' design for similar games.

No session lasted more than two hours. Players could earn a maximum of \$41.50 in the ACA' sessions and \$51.00 in the ABA' sessions. This difference reflects the fact that the produrally rational algorithm never makes an mistake and, hence, it is not possible to win balanced games in treatment C. In order to earn the maximum of \$51 a subject had to win not only all unbalanced games in all three treatments, but also the balanced games in treatment B. In the ACA' sessions, the subject could theoretically earn slightly more than \$41.50 if the computer timed out and thus lost an unbalanced game in treatment C.

4 Experimental Results

Earnings varied greatly across subjects. Subject earnings ranged from \$14.00 to \$39.00. The average earnings were \$23.96, 58 percent of the earnings that substantively rational players would earn. The average subject in the ABA' sessions earned \$27.39 and the average subject in the ACA' design earned \$20.54 Subjects with the highest earnings were from the ABA' sessions. The procedurally rational algorithm in the ACA' sessions was rarely defeated even when it started in a losing position, because the middle treatment games are difficult to solve by backwards induction.

The average subject won 34 percent of the unbalanced games, which have a game theoretic value of a win, when in teh role of the first mover. In treatments A and A', 9 of the 31 games were balanced and thus impossible for the subject to win. In treatments B and C half of the games, 19, were balanced, which means that when playing against the procedurely rational algorithm in treatment C they were unwinnable.

Figures 3(a) and 3(b) plot the win frequencies by game for the A and A' treatments by session design. The games are ordered by the probability a blunderer would win them starting on the left with (1,0,0,1,1), which can not be lost, and ending with (4,0,2,7,0), which almost no one wins even though it is a winnable game. The subjects do better in the simpler games like (2,1,0,0,1) and (0,0,0,0,6) where the win frequency approachs 100 percent. In the simple cases on the left of the figures, the subjects' win frequency is significantly greater than the blunderer model would predict. However, as the games get more complicated, human behavior in only a few games is statistically different from blunderer behavior. These spikes are games that allow the move copying heuristic. For example, even though (0,6,0,2,6) is a rank 14 game it is easy to win if you know the move copying heuristic. Simply, take the two stones in the forth row leaving (0,6,0,0,6) and move copy until the end of the game and the human wins. As the games get more complicated, human performance in the games where the move copying heuristic is applicable is considerably greater than in the games without a solution by the move copying heuristic.

Fig. 2: Treatment *B* Graphical User Interface

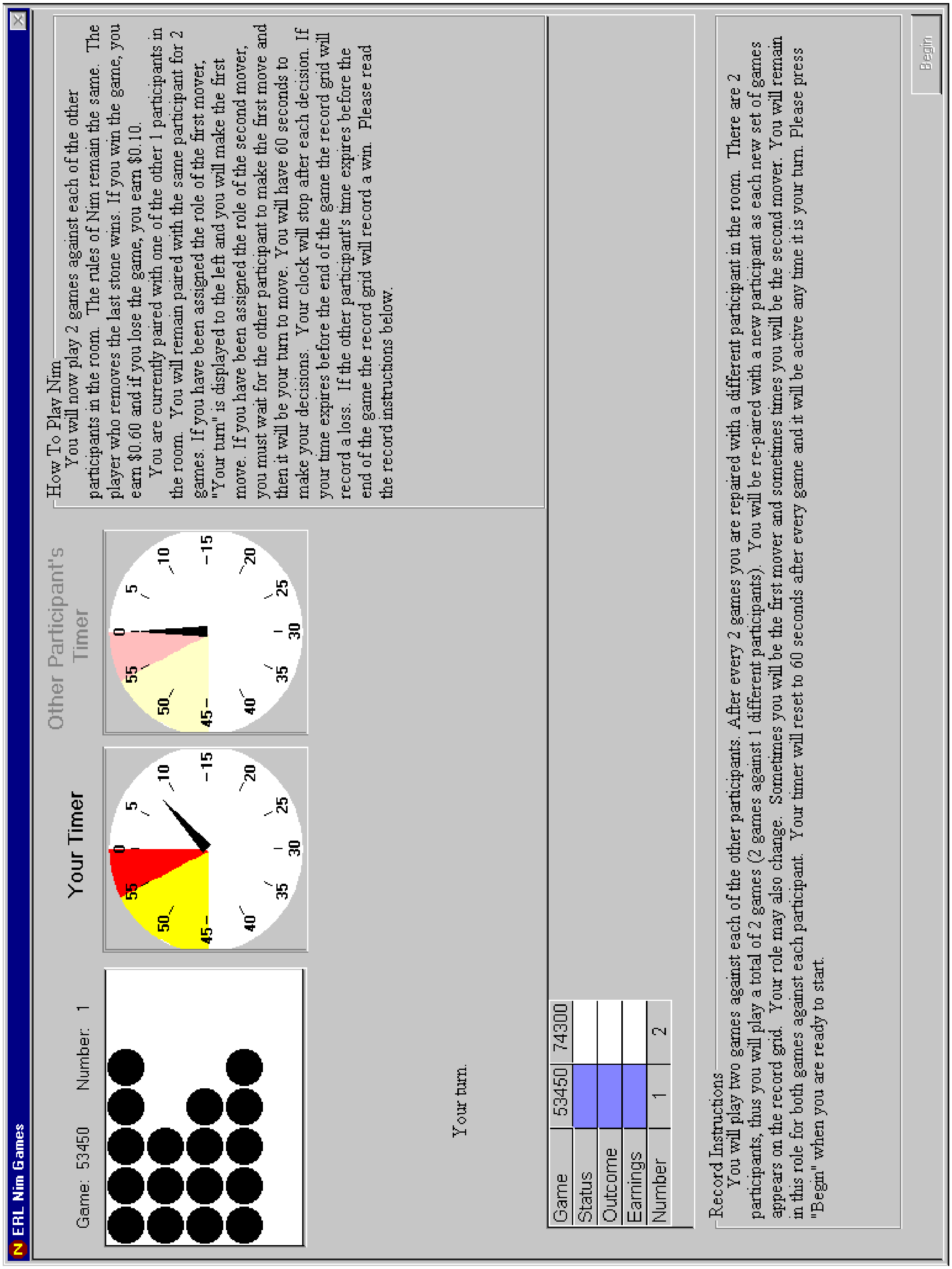
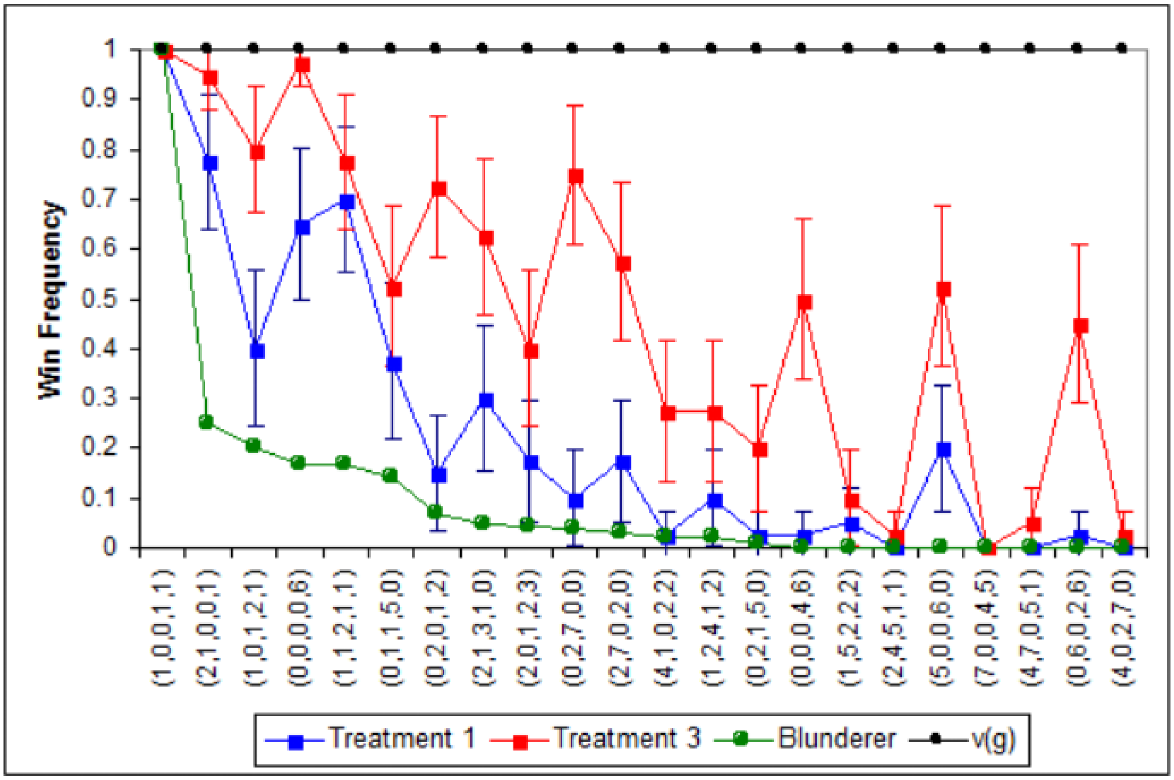
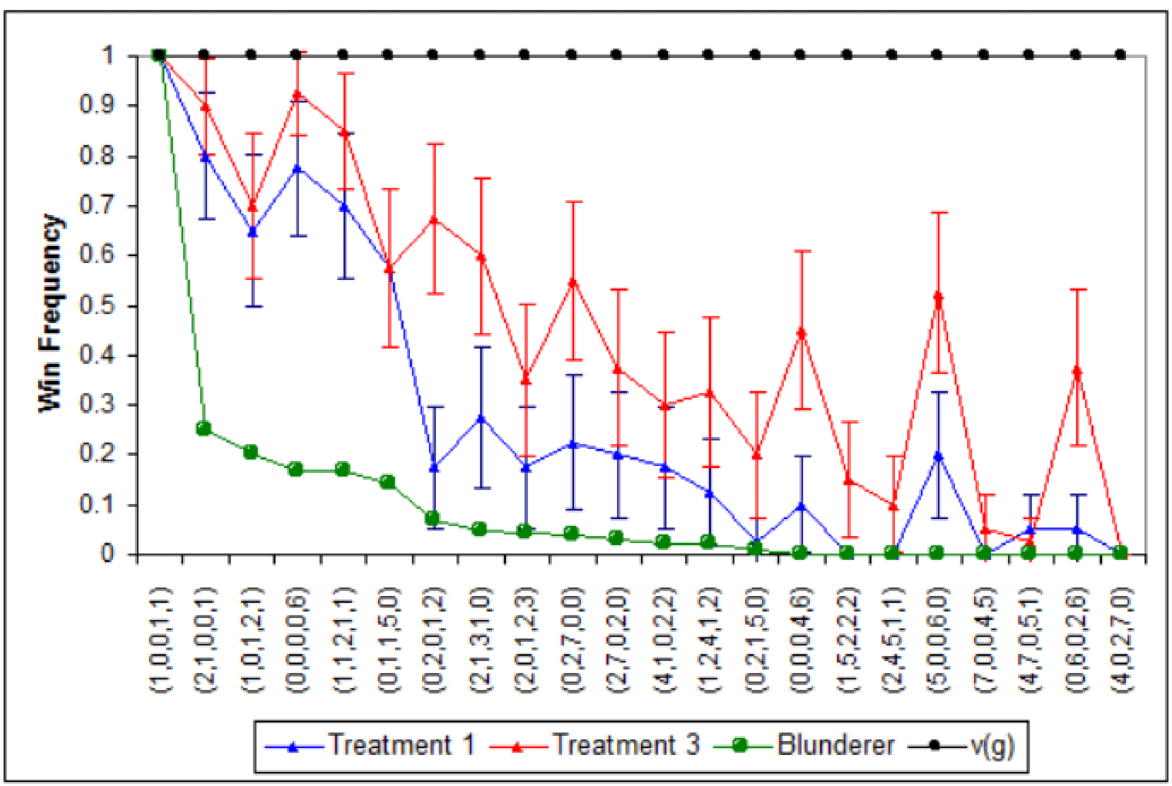


Fig. 3: Win frequencies by game

(a) ABA' sessions



(b) ACA' sessions



Another feature of Figures 3(a) and 3(b) is that the red line connecting performance in treatment $A'$is consistently above the blue line connecting performance in treatment A. There is economically significant learning going on between the two treatments.

Paired comparisons of the average win percentages for all unbalanced games in treatments A and A' are presented in Table 1. In all cases there is an increase in win percentage across the two treatments. T-tests confirm the differences in the means and non-parametric tests show that the distributions are not the same. P-values for all the tests are presented in the last three columns of the table. All test statistics were significant at the 99 percent level. On average the subjects do statistically and economically significantly better in treatment A' than in treatment A.

There is more learning in the sessions played against a human, ABA', than in the sessions played against the procedurally rational algorithm, ACA'. Against humans the increased win percentage is 31 percent in move copying games. In the ACA' sessions the improvement is 21 percent. For non-move copying games the improvement is small, 9 percent and the same for both. Playing 38 games with a human teaches our subjects how to win winnable move copying games more effectively than playing the 38 games against a procedurally rational algorithm.

Although the comparisons of mean win percentages provide strong evidence of learning, the heterogeneous payoffs across subjects suggest that there are differences that are not detectible in the means. In order to account for changes at the subject level we developed a random effect logit model. We model the probability of winning a game in Treatments A and A' as a function of rank, treatment (A or A'), design (ABA' or ACA') , and the presence of the move copying heuristic (MC or nonMC). We also interact the treatment and MC variables. The baseline is the nonMC games in treatment A. We then add the interactions to detect differences across the ABA' and ACA' designs. The results of the model are presented in Table 2.

The coefficients for Rank, Treatment, and Move Copying in treatment A' are statistically significant at the 95 percent level or better. Focusing on the coefficient for rank in the baseline case gives an estimated rationality bound of rank 5, that is, subjects win about half the games of rank 5. This is similar to, but slightly less than, the measured rationality bounds reported in McKinney and Van Huyck [2007].[5] This may be a more accurate measure as it accounts for the move copying heuristic. Notice that all rank 5 Nim games are winnable and can be won by using the move copying heuristic. The measured rationality bound is again found to be remarkably low.

The model affirms that subjects have more trouble solving the higher ranked games in the first treatment in which the move copying heuristic is not applicable. The model also interacts the treatment and move copying variables with the ABA' dummy variable. There is no statistical difference in the ABA' subjects' ability to solve most of the games, but subjects do significantly better in move copying games in treatment A'. The random effects logit model confirms the economic and statistical significance of the learning that takes place in treatment B. When learning the move copying heuristic, it helps to play against an error prone opponent.

5 Conclusion

The ability to recognize the move copying heuristic plays a major role in a subject's ability to win winnable Nim games. In all treatments our subjects perform significantly better in games where the move copying heuristic is applicable. They learn the move copying heuristic more effectively when playing against error prone opponents like themselves. We found that subjects that had experience playing games against other humans showed a significant increase in their ability to solve games in which the move copying heuristic is applicable.

[5] Our previous studies did not limit the time subjects had to play the game. Being timed may account for the lower estimated rationality bound. Deck and Jahedi [2013] show that cognitive load can influence economic decision making.

References

Charles L. Bouton. Nim - a game with a complete mathematical theory. *The Annals of Mathematics*, III:35–39, 1901-1902.

J.H. Conway. *On Numbers and Games*. A.K. Peters, Ltd, 2nd edition, 2001.

Cary Deck and Salar Jahedi. The effect of cognitive load on economic decision making. 2013.

Martin Dufwenberg, Ramya Sundaram, and David J. Butler. Epiphany in the game of 21. *Journal of Economic Behavior and Organization*, 75:132–143, 2010.

Uri Gneezy, Aldo Rustichini, and Alexander Vostroknutov. Experience and insight in the race game. *Journal of Economic Behavior and Organization*, 75:144–155, 2010.

Carl N. McKinney, Jr. and John Van Huyck. Does seeing deeper into a game actually increase one's chance of winning? *Experimental Economics*, pages 297–303, 2006.

Carl N. McKinney, Jr. and John Van Huyck. Estimating bounded rationality and pricing performance uncertainty. *The Journal of Economic Behavior and Organization*, pages 625–639, 2007.

Carl N. McKinney, Jr. and John Van Huyck. Eureka learning: Heuristics and deliberation time in perfect information games. *Games and Economic Behavior*, pages 223–232, 2013.

Table 1: Win percentage for Unbalanced Games-Paired tests

		Number of games	Treatment A	Treatment A’	Difference	t-tests Ho: Equal means	Sign tests Ho: Equal medians	Smirnov tests Ho: Equal distributions
ABA	**MC**	15	30.17%	61.00%	30.83%	0.000	0.0000	0.0000
	nonMC	7	10.36%	19.64%	9.29%	0.000	0.0000	0.0000
	All	22	23.86%	47.84%	23.98%	0.0000	0.0000	0.0000
ACA	**MC**	15	36.17%	57.00%	20.83	0.000	0.0001	0.0000
	nonMC	7	12.14%	20.71%	8.57%	0.0005	0.0025	0.0000
	ACA	22	28.52%	45.45%	16.93%	0.0000	0.0001	0.0000
Total	**MC**	15	31.17%	59.00%	25.83%	0.000	0.0000	0.0000
	nonMC	7	11.25%	20.18%	8.93%	0.000	0.0000	0.0000
	Total	22	26.19%	46.65%	20.45%	0.0000	0.0000	0.0000

The p-values are reported for all statistical tests in the last three columns. All tests are calculated using sample sizes of n=40 subjects in each design. MC denotes move copying games and nonMC denotes games that are not move copying games.

Table 2: Random Effects Logit Model of the Probability of Winning

	Variable	Coefficient	Standard Error
	Constant	2.284**	.268
	Rank	-0.462**	.019
	MC games in Treatment A	0.361	0.207
	MC games in Treatment A'	1.771**	0.206
	nonMC games in Treatment A	*Baseline*	
	nonMC games in Treatment A'	1.023**	0.244
Interaction with ABA	*MC games in Treatment A*	-0.085	0.290
Interaction with ABA	*MC games in Treatment A'*	0.613*	0.289
Interaction with ABA	*nonMC games in Treatment A*	0.218	0.430
Interaction with ABA	*nonMC games in Treatment A'*	0.287	0.382
	n	3520	
	subjects	80	
	games	44	
	log likelihood	-1527.47	

***99%, *95%,*

APPENDIX A

Treatment 1 Games

Game *g*	Value *v(g)*	Rank *r(g)*	Rows *m*	Log of NT-complexity	Blunderer's win prob.
(1,0,0,1,1)	W	3	3	-	1.0000
(2,1,0,0,1)	W	4	3	1.61	0.2500
(2,2,0,0,0)	L	4	2	1.95	0.0000
(1,0,1,2,1)	W	5	4	2.77	0.2000
(0,0,0,0,6)	W	6	1	2.77	0.1667
(0,2,0,1,2)	W	5	3	3.09	0.0667
(1,1,2,1,1)	W	6	5	4.17	0.1667
(3,1,2,0,0)	L	6	3	4.23	0.0000
(0,1,1,5,0)	W	7	3	4.82	0.1429
(2,1,3,1,0)	W	7	4	5.79	0.0476
(0,2,1,5,0)	W	8	3	6.33	0.0083
(0,2,7,0,0)	W	9	2	6.44	0.0370
(1,0,3,3,1)	L	8	4	7.16	0.0000
(2,0,1,2,3)	W	8	4	7.39	0.0417
(0,0,0,4,6)	W	10	2	8.00	0.0010
(4,1,0,2,2)	W	9	4	8.64	0.0222
(5,0,0,6,0)	W	11	2	9.11	0.0001
(1,4,4,1,0)	L	10	4	9.71	0.0000
(2,7,0,2,0)	W	11	3	9.93	0.0303
(0,2,3,3,2)	L	10	4	10.48	0.0000
(1,2,4,1,2)	W	10	5	10.63	0.0200
(0,6,0,4,2)	L	12	3	11.70	0.0000
(4,0,2,7,0)	W	13	3	12.80	0.0000
(1,4,1,5,1)	L	12	5	13.10	0.0000
(1,5,2,2,2)	W	12	5	13.74	0.0008
(0,6,0,2,6)	W	14	3	14.18	0.0000
(0,3,6,5,0)	L	14	3	14.57	0.0000
(2,4,5,1,1)	W	13	5	14.98	0.0001
(7,0,0,4,5)	W	16	3	17.26	0.0001
(4,7,0,5,1)	W	17	4	19.77	0.0000
(3,3,1,5,4)	L	16	5	20.42	0.0000

Treatment 2 Games

Game *g*	Round	Value *v(g)*	Rank *r(g)*	Rows *m*	Log of NT-complexity	Blunderer's win prob.
(5,3,4,5,0)	1	0	17	4	20.97	0.0000
(7,4,3,0,0)	1	1	14	3	14.39	0.0000
(5,4,0,4,4)	2	0	17	4	21.11	0.0000
(2,0,5,3,4)	2	1	14	4	16.21	0.0000
(7,0,3,7,0)	3	0	17	3	18.27	0.0000
(2,5,0,5,2)	3	1	14	4	15.92	0.0000
(6,3,7,1,0)	4	0	17	4	19.54	0.0000
(1,5,1,0,5)	4	1	12	4	12.19	0.0000
(3,0,2,7,5)	5	0	17	4	20.23	0.0000
(0,7,2,4,1)	5	1	14	4	15.09	0.0000
(4,7,2,4,0)	6	0	17	4	20.36	0.0000
(6,0,7,0,1)	6	1	14	3	13.46	0.0000
(6,4,5,2,0)	7	0	17	4	20.55	0.0000
(3,3,3,0,3)	7	1	12	4	13.56	0.0000
(0,6,5,6,0)	8	0	17	3	18.77	0.0000
(3,0,4,6,1)	8	1	14	4	15.56	0.0000
(6,0,4,3,4)	9	0	17	4	20.87	0.0000
(0,6,6,0,0)	9	1	12	2	10.19	0.0000
(5,3,0,3,6)	10	0	17	4	20.73	0.0000
(2,7,0,0,5)	10	1	14	3	14.10	0.0000
(4,6,1,0,6)	11	0	17	4	19.86	0.0000
(0,4,4,2,2)	11	1	12	4	13.24	0.0000
(2,0,2,7,6)	12	0	17	4	19.81	0.0000
(4,3,3,4,0)	12	1	14	4	16.51	0.0000
(5,0,5,5,2)	13	0	17	4	20.66	0.0000
(1,5,6,2,0)	13	1	14	4	15.27	0.0000
(6,0,6,3,2)	14	0	17	4	20.31	0.0000
(1,6,6,0,1)	14	1	14	4	14.62	0.0000
(2,7,1,0,7)	15	0	17	4	19.03	0.0000
(7,0,0,0,7)	15	1	14	2	12.31	0.0000
(4,6,1,5,1)	16	0	17	5	21.05	0.0001
(3,4,4,1,2)	16	1	14	5	17.21	0.0000
(4,7,1,2,3)	17	0	17	5	21.31	0.0000
(1,1,6,2,4)	17	1	14	5	16.26	0.0000
(2,2,5,3,5)	18	0	17	5	22.20	0.0000
(3,2,1,3,3)	18	1	12	5	14.19	0.0000
(2,2,5,7,1)	19	0	17	5	21.00	0.0000
(1,5,2,2,4)	19	1	14	5	21.00	0.0000