A Dynamic ACT-R Model of Simple Games

Christian Lebiere (cl+@cmu.edu)
Psychology Department
Carnegie Mellon University
Pittsburgh, PA 15213

Robert L. West (rwest@hku.hk)
Department of Psychology
Carleton University
Ottawa, Canada

Abstract

A model of humans playing the simple game of Paper Rock Scissors based on the ACT-R architecture (Anderson, 1993; Anderson & Lebiere, 1998) is presented. This model stores in long-term memory sequences of moves and attempts to anticipate the opponent's moves by retrieving from memory the most active sequence. This results in a tightly linked dynamical system in which each player drives the play of its opponent. The performance of this model as a function of the length of the sequences stored and the amount of noise in the system is investigated, and is compared to the performance of human subjects.

Introduction

From the point of view of classical game theory (e.g. von Neumann & Morgenstern, 1944; Nash, 1950; Fudenberg & Tirole, 1991), the simple game of Paper Rock Scissors (PRS) is quite trivial. Each of the three possible moves is as good as the other ones: Paper beats Rock, Rock beats Scissors and Scissors beats Paper. Since the players make their moves simultaneously without any a priori knowledge of each other's move, the optimal strategy is to play randomly and thus guarantee the expected outcome of a tie. However, it is generally accepted that game theory's optimally rational strategies often do not accurately describe human behavior due to the fact that human rationality is bounded (Simon, 1972). Also, game theory does not provide an account of how human players learn. Instead, human game players are best viewed as cognitively limited learners (Erev & Roth, 1998).

As Bracht, Lebiere and Wallach (1998) have demonstrated, ACT-R can be used to model how strategies are applied by conceptualizing the possible moves as productions. There are two advantages to this approach. The first is that ACT-R has been used to model many behavioral phenomena and thus it integrates game playing into the larger context of human cognition. The second is that the method for selecting between productions is consistent with the way game playing is understood in game theory and in Experimental Economics. That is, each move is associated with a probability that reflects its utility. Thus, while game theory can provide the optimal distribution of the probabilities, ACT-R can provide a cognitively justifiable account of how the actual probabilities are learned.

However, recently West (1998a, 1998b, 1999) has provided an alternative, dynamic systems account of how simple games are played, based on the principle of *reciprocal causation*. Reciprocal causation refers to a state in which two systems are coupled together so that each system's outputs are affected by the other system's behavior (Clark, 1997, 1998). The importance of reciprocal causation is that it is often associated with "emergent behaviors whose quality and complexity far exceeds that which either subsystem could display in isolation" (Clark, 1998). The approach of West (1998a, 1998b, 1999) is based on the findings that humans are quite bad at generating random outputs (see Tune, 1964, and Wagenaar, 1972 for reviews), but quite good at detecting sequential dependencies (e.g. Ward, 1973; Ward, Livingston, & Li, 1988). West (1998a, 1998b, 1999) assumed that players attempt to predict their opponent's next move by detecting sequential dependencies in their opponent's past moves and modeled the process using neural networks. The result was that the modeled players were in a state of reciprocal causation, i.e. each player's moves were determined by their opponent's previous moves.

The reciprocal causation resulted in a chaos-like process that caused both players to generate outputs that appeared random. This result was consistent with the game theory prediction but it was contingent on the players being evenly matched in terms of how many previous moves (lags) they could remember (it was assumed that the players could only remember a limited number of lags back on each trial). When the players were unequally matched in terms of how many lags back they could remember, the player who could remember more lags enjoyed a systematic advantage. Importantly, this was also found to be the case for human subjects (West, 1998a, 1998b, 1999).

This phenomena, which can be considered an emergent property of the dynamic interaction between the players, is very difficult to account for by treating the moves as productions with associated, learned utility values. However, unlike the various specialized models in Experimental Economics, ACT-R is not limited to learning

in this way. As Lebiere and Wallach (1998) have demonstrated, the declarative memory system of ACT-R can be used to account for implicit learning tasks without relying on production-based learning. In this paper we demonstrate how the neural network-like qualities of the ACT-R declarative memory system can produce the same phenomena found by West (1998a, 1998b, 1999) in a very straightforward manner.

Model

To emulate the neural network model of West (1998a, 1998b, 1999), we used a simplified version of the ACT-R Sequence Learning Model (Lebiere & Wallach, 1998) that operated by building chunks encoding short sequences of stimuli. For clarity, if the model builds sequences of moves of length 3, we will call it a lag2 model because it remembers the previous two moves of the opponent in addition to its current move. Similarly, a lag1 model refers to sequences of length 2. We will describe below a lag2 model, but we will also report results for a lag1 model.

ACT-R is a goal-directed architecture. At all times, the system focuses on a single goal, and any production must first match that goal before firing. In this model, the current goal can be understood as the player's working memory (Lovett, Reder & Lebiere, in press). It holds a number of the opponent's previous moves in a chunk such as:

Goal
isa PRS
lag2 Paper
lag1 Rock
lag0 nil

PRS is the *type* of the goal, and its *slots* are lag2, lag1 and lag0[1]. Lag0 holds the opponent's current move (a value of nil indicates that that move has not yet been played), lag1 holds the opponent's previous move (Rock) and lag2 holds the opponent's move before that (Paper). After a move is made and the lag0 value is filled in, the goal is popped and becomes a chunk in declarative memory. If an identical chunk already exists, then that chunk is reinforced instead of creating a copy.

The model is composed of three productions. The main production, **Sequence Prediction**, attempts to retrieve from memory a chunk that encodes a sequence of three moves (L2, L1, L) played by the opponent, the first two of which match the opponent's last two moves (L2, L1). Then given the third move of that sequence (L), it retrieves the move that beats it (M) and plays that move (M).

Sequence Prediction
IF no move has been played
  and the opponent last played moves L2 and L1
  and moves L2 and L1 are usually followed by move L
  and move L is beaten by move M
THEN play move M

This corresponds to trying to anticipate the move that the opponent is going to make given his most recent moves and making the move that defeats it. If no such sequence of the opponent's moves can be retrieved from memory (for example, at the start of the game), then the second production, **Random Guess**, applies. It simply selects a move (L) at random and plays the move that defeats it (M).

Random Guess
IF no move has been played
  and move L is beaten by move M
THEN play move M

Finally, after the players have each made their move, the third production, **Next Move**, applies. It records the opponent's move (L) in the current goal, thus completing the opponent's most recent three-move sequence (L2, L1, L). It then pops that goal, which becomes a chunk in declarative memory (or reinforces an identical chunk if it already exists), and focuses on a new goal which contains the opponent's two most recent moves (L1, L).

Next Move
IF the opponent has played move L after moves L2 and L1
THEN note move L in the current goal, pop that goal and
  focus on a new goal holding previous moves L1 and L

The production cycle can then start anew. The crucial part of this model is the retrieval from long-term declarative memory in production **Sequence Prediction** of a chunk holding the opponent's move sequence matching the current situation. Retrieval from memory depends upon a chunk's activation. Anderson and Schooler (1991) reported that the odds of an item in the environment being needed decrease as a power function of its past uses. In ACT-R, the activation of a chunk[2] is interpreted as the logarithm of the odds of that chunk being needed from memory, and thus will be defined as:

$$A_i = \ln \sum_{j=1}^{n} t_j^{-d} \qquad (1)$$

A_i is the activation of chunk i, n is the total number of past references to that chunk, t_j is the time since the jth reference and d is the decay rate. This activation equation incorporates both the power law of practice (through the summation) and power law decay (of each reference). Past references refer both to chunk creation (and re-creations) and to retrievals from memory. If the references are assumed to be evenly distributed over the chunk's past history, then the activation of the chunk can be simplified to be[3]:

[1] PRS, lag2, lag1 and lag0 are arbitrary names to designate the goal type and its slots.

[2] Strictly speaking, this is only the base-level activation. Additional components of activation include spreading activation and mismatch penalties, but neither is relevant to this model.

[3] For efficiency reasons, the results reported in the next section correspond to models for which Equation (2) is used instead of

$$A_i = \ln \frac{n \cdot L^{-d}}{1-d} \qquad (2)$$

L is the life of the chunk, i.e. the length of time since its creation (t_j). If several chunks satisfy the condition, then the one with the highest activation is retrieved. Zero-mean Gaussian noise is added to the activations, which makes retrieval a probabilistic process. The probability of retrieving chunk i among all the alternatives j is a function of their respective activations and the magnitude of the noise:

$$p(i) = \frac{e^{A_i/t}}{\sum_j e^{A_j/t}} \qquad (3)$$

t is a measure of the noise proportional to its standard deviation[4]. Assuming that all the chunks were created around the same time, i.e. have a similar L, then Equations (2) ad (3) can be simplified to yield:

$$p(i) = \frac{n_i^{1/t}}{\sum_j n_j^{1/t}} \qquad (4)$$

A noise value t of 1 would yield Luce's linear choice rule (Luce, 1959). As Lebiere (1998) established, the noise magnitude t is the crucial parameter that determines the dynamics of the retrieval process. When t=1, the probabilities of retrieval match the distribution of past references[5], and retrieval leaves the statistics of occurrence unchanged. For values of t larger than 1, the differences in past references are reduced, and retrieval becomes increasingly random. For values of t smaller than 1, the system becomes increasingly deterministic in selecting the most active chunk. A rich-get-richer dynamics develops, in which the most active chunks become even more so and the less active ones gradually decay away.

Essentially, the model uses the declarative memory system of ACT-R to detect sequential dependencies. Of course, this is only the behavior of a single cognitive system in isolation. Similar to West (1998a, 1998b, 1999), when two of these systems are coupled together the result is a state of reciprocal causation. Thus the important question was whether this particular coupled system would produce the same emergent pattern of behavior that West (1998a, 1998b, 1999) found in his models and human subjects.

Equation (1). There was little indication however that the simplification altered in any way the behavior of the model.

[4] Formally, $t = \sqrt{6}\sigma/\pi$ where σ is the standard deviation.

[5] The phenomenon of reproducing in one's choices the probabilities of occurrence of events in the environment is known as probability-matching (Friedman et al., 1964; Myers, Fort, Katz, and Suydam, 1963).

Results

We will first describe the behavior of the model playing against an identical copy of itself. The model is a lag2 model as described in the previous section. The only parameter is the noise magnitude of 0.25. This parameter is taken from the model of (Lebiere, 1998), which reflected stochasticity in the learning of arithmetic. Examining the difference in score across trials, the output resembles a random walk, with possible fractal properties.

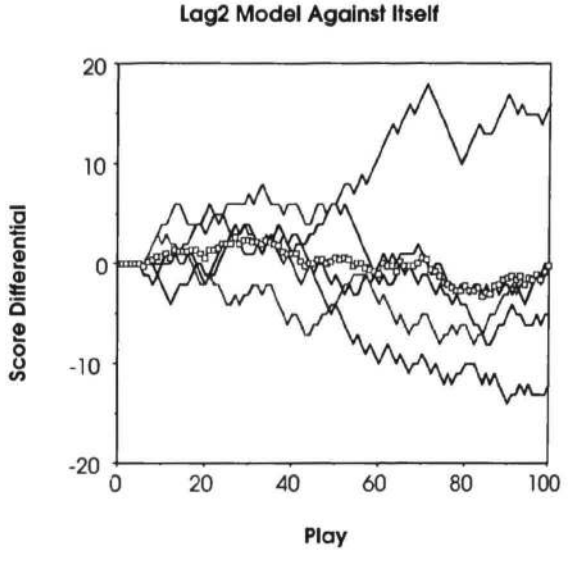

Figure 1: Score differential of lag2 model vs. itself. 5 sample runs of 100 plays. Mean of the 5 runs in squares.

The next question was whether an imbalance between the players in terms of working memory would produce a bias in favor of the player who processed more lags.

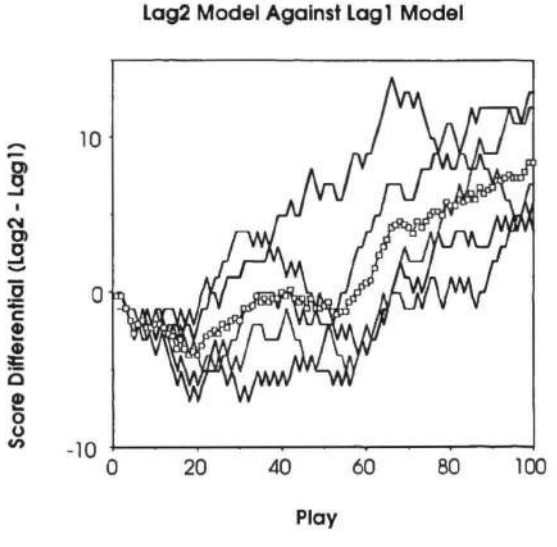

Figure 2: Score differential of lag2 model vs. lag1 model. 5 sample runs of 100 plays. Mean of the 5 runs in squares.

While the differential in score between the lag 2 and lag1 models fluctuates as it did between evenly matched models, the long-term trend is clearly in favor of the more powerful lag2 model. But how do these models compare to humans? West (1998a, 1998b, 1999) found that humans play similarly to a lag2 model in that they are able to beat a lag1 model. Following this approach we had human subjects play against the ACT-R lag1 model. The subjects were five participants in the ACT-R summer school.[6]

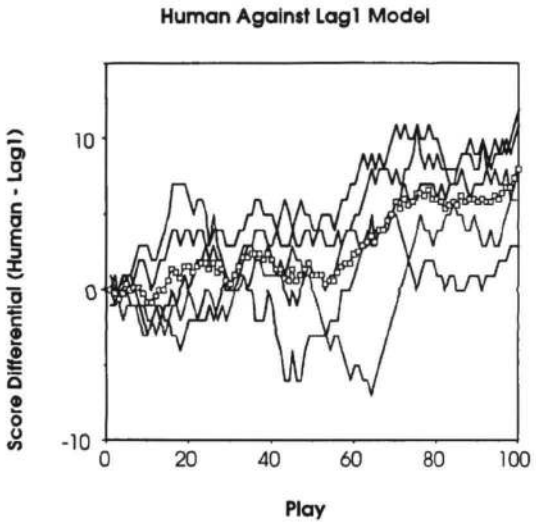

Figure 3: Score differential of humans vs. lag1 model. 5 sample runs of 100 plays. Mean of the 5 runs in squares.

The results were very similar to West (1998a, 1998b, 1999) and also to the performance of the lag2 model playing against the lag1 model (Figure 2), including the fluctuations in the score differential and the average winning margin against the lag1 model. An intriguing feature is that in both Figures 2 and 3 the superior (lag2) player initially loses against the lag1 model, then somewhere between approximately 20 and 30 trials begins to win. This is consistent with the fact that the lag2 model builds longer chunks than the lag1 model, and thus takes longer to accumulate the proper set of sequences. Thus in this range the prediction is reversed and the lag1 model should perform better than the lag2 model. To test this prediction we had 8 human subjects from the University of Hong Kong play short games of 30 trials each against both a lag1 model and a lag2 model. The results, displayed in Figure 4, show that early on the lag1 model is indeed more difficult to beat than the lag2 model. A paired t-test on the score differences revealed that this difference was significant at $P<.001$.

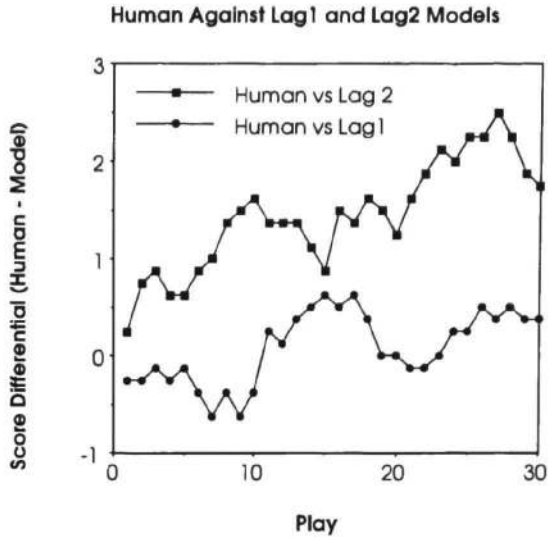

Figure 4: Score differential of humans vs. lag1 and lag2 models for 30 plays. Mean of eight subject runs.

While a larger lag is clearly an advantage, what about the other variable characteristic of our model, the noise? If two models have the same lag and identical noise levels, then as we have seen they will play evenly in the long run. If one model has a very high noise level, it will play randomly (the game theory solution) and will also draw in the long run. This can be a good thing if a player is intrinsically at a disadvantage, as when a lag1 model plays a lag2 model. But is randomness simply a way for a player to limit its losses against a superior opponent? What if both networks have the same lag but different limited noise levels? Is noise an advantage or a disadvantage? Obviously the noisier model is less predictable, but it is also a less powerful learner, slower to pick up on existing sequential dependencies.

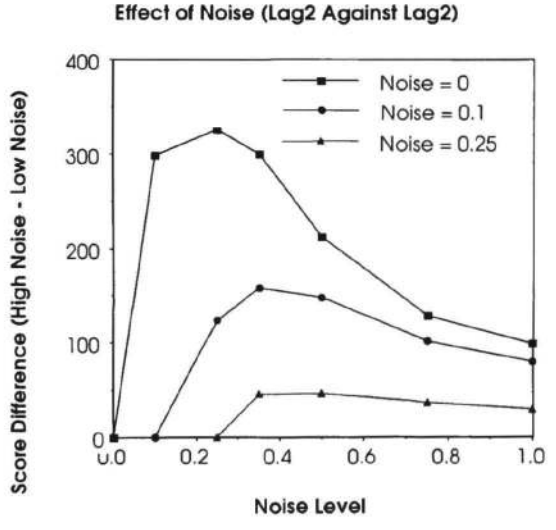

Figure 5: Average over 200 runs of 1000 plays of the final difference in score between two lag2 models with different noise levels.

[6] The model is available for playing on the world-wide web at http://bk1.psy.cmu.edu/inter/models?

We see that all things being equal a higher noise level is indeed an advantage. The advantage in differential score increases for a while as the difference in noise levels increases, then declines because the whole system just becomes increasingly random. To further investigate, we ran a lag2 model against a lag1 model at various noise levels.

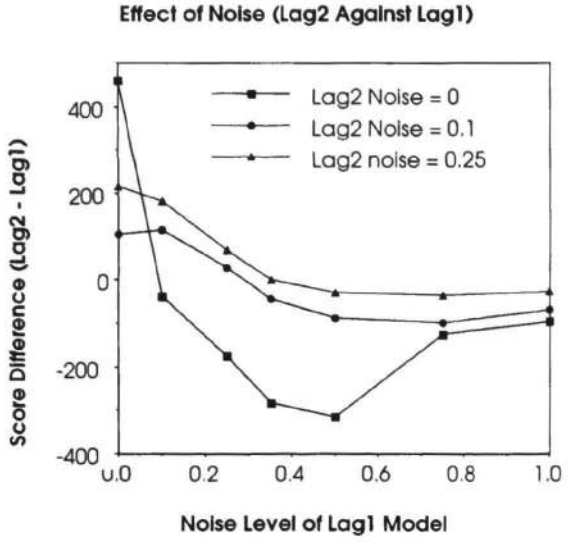

Figure 6: Average over 100 runs of 1000 plays of the final difference in score between a lag2 model and a lag1 model with different noise levels.

The results show that noise can override the lag factor causing a lag1 model to beat a lag2 model. This was particularly true when the noise level of the lag2 model was set to zero or was very low.

Discussion

By manipulating parameters of the model we were able to recreate the phenomena found by West (1998a, 1998b, 1999) and generate further predictions (some of which have yet to be confirmed with human subjects). It is tempting to interpret our results in terms of the individual models. For example, in terms of our findings for number of lags processed and the amount of noise, one interpretation is that the beneficial effect of behaving less predictably (more noise) can outweigh the effects of being a more powerful learner (more lags). However, the picture is potentially more complex. When a model wins it does so by predicting its opponent's moves from sequential dependencies present in past behavior. The past behavior is generated by a reciprocal causation process between the models that results in a chaos-like process. This process produces sequential dependencies as well as a random walk quality. The problem is that it is very difficult to disentangle the process that generates the outputs from the ability of the winning network to detect sequential dependencies, since the action of detecting sequential dependencies is also part of the process that generates them. If we change the way the models detect sequential dependencies we also change the way the sequential dependencies are generated. Thus, we cannot, at this point, rule out the possibility that the noise factor or the lag factor may operate by altering the type of sequential dependencies produced by the system.

The noise factor can also be understood in another way. Lebiere (1998) found that randomness serves a beneficial cognitive function by keeping the system's dynamics fluid and thus preventing errors from becoming entrenched facts. In our model the opponent is always changing in response to the history of the game. Facts are a short-lived phenomenon in this constantly changing environment. Thus an injection of stochasticity may have the effect of optimizing the system for this environment. More generally, this type of environment probably provides a much better replica of the environment in which our cognitive system evolved than a formal system of unchanging facts and rules such as arithmetic.

The ACT-R model that we used has many features in common with the neural network model of West (1998a, 1998b. 1999). They both work by storing sequences of the opponent's moves and using them to anticipate the opponent's next move. West's (1998a, 1998b, 1999) fixed-length two-layer feedforward neural networks, whose inputs are the opponent's past moves and whose output is the opponent's next move, correspond closely to the ACT-R model's chunks, whose slots holding the opponent's past moves are primed during memory retrieval and whose output is the value of the slot holding the next move. Also, both models resort to a random choice in the case of two moves being equally weighted. However, (again in both models) the main source of randomness is the chaos-like effect generated by coupling the networks together in a state of reciprocal causation. This effect is also the source of the sequential dependencies, which would not occur if the process were based on a truly random process.

However, the ACT-R model has several advantages. First, because ACT-R is a unified cognitive architecture, the model is more informative as to the cognitive structures involved in the process. Specifically, the ACT-R model situates the detection of sequential dependencies in declarative memory, while the lag factor can be interpreted in terms of the amount of working memory (Lovett, Reder & Lebiere, in press). ACT-R also allows for a principled investigation of background random noise, which turns out to be an important factor. Also, ACT-R is capable of modeling more complex games, involving knowledge and strategy, through the use of productions. Because these games also often involve an element of guessing, we suggest that a full model of game playing will integrate both processes. ACT-R is important in this respect because it provides a ready-made model of how to structure this integration.

The origins of this ACT-R model should be emphasized to illustrate the lack of degrees of freedom in its conception. The basic idea to play PRS by storing fixed-length sequences of the opponent's moves was adopted from West (1998a, 1998b, 1999), as was the default length of those sequences. The very simple chunks and productions used to implement that idea were taken from an existing ACT-R model of a seemingly very different paradigm from another

field. The default value of the only parameter, the magnitude of the noise, came from an ACT-R model of a separate phenomenon. Those elements were assembled almost automatically and provided a very accurate model of human playing. That it happened on the first try, without any engineering or parameter tuning, is a demonstration of the predictive power of unified architectures.

Conclusion

We proposed a model of human playing for Paper Rock Scissors. This model was inspired by known psychological limitations and inclinations instead of the ideal strategies of classical game theory. The players were viewed not as isolated cognitive entities but as part of a dynamical system in which they constantly influence each other's actions. Crucial parameters of this cognitive system are the raw power of the actors in terms of the length of sequences that the players can process and the degree of stochasticity with which they select their actions. This model was found to closely account for human behavior, without the benefit of unexamined degrees of freedom in its knowledge structures or parameters.

Acknowledgements

This research was partially supported by a grant from the Office of Naval Research (N00014-95-10223).

References

Anderson, J. R. (1993). *Rules of the Mind.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. R., & Lebiere, C. (1998). *The Atomic Components of Thought.* Mahwah, NJ: Lawrence Erlbaum Associates.

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, 2, 396-408.

Bracht, J., Lebiere, C., & Wallach, D. (1998). On the need of cognitive game theory: ACT-R in experimental games with unique mixed strategy equilibria. Paper presented at the Joint Meetings of the Public Choice Society and the Economic Science Association, New Orleans, LA.

Clark, A. (1997). *Being there: Putting brain, body and world together again.* Cambridge, MA: MIT Press.

Clark, A. (1998). The dynamic challenge. *Cognitive Science*, 21 (4), 461-481.

Erev, I., & Roth, A. E. (1998). Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. *American Economic Review*, 88(4), 848-881.

Friedman, M. P., Burke, C. J., Cole, M., Keller, L., Millward, R. B., & Estes, W. K. (1964). Two-choice behavior under extended training with shifting probabilities of reinforcement. In R. C. Atkinson (Ed.), *Studies in Mathematical Psychology* (pp. 250-316). Stanford, CA: Stanford University Press.

Fudenberg, D., & Tirole, J. (1991). *Game Theory.* Cambridge, MA: MIT Press.

Lebiere, C. (1998). The dynamics of cognition: An ACT-R model of cognitive arithmetic. Ph.D. Dissertation. *CMU Computer Science Dept Technical Report* CMU-CS-98-186. Pittsburgh, PA.

Lebiere, C. & Wallach, D. (1998). Implicit does not imply procedural: A declarative theory of sequence learning. Paper presented at the *41st Conference of the German Psychological Association*, Dresden, Germany.

Lovett, M. C., Reder, L. M., & Lebiere, C. (in press). Modeling working memory in a unified architecture: An ACT-R perspective. To appear in Miyake, A. & Shah, P. (Eds.) *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control.* New York: Cambridge University Press.

Luce, R. D. (1959). *Individual Choice Behavior: A Theoretical Analysis.* New York: Wiley.

Myers, J. L., Fort, J. G., Katz, L., & Suydam, M. M. (1963). Differential monetary gains and losses and event probability in a two-choice situation. *Journal of Experimental Psychology*, 66, 521-522.

Nash, J. (1950). Equilibrium points in N-person games. *Proceedings of the National Academy of Sciences*, 36, 48-49.

Simon, H. A. (1972). Theories of bounded rationality. In C. B. Radner & R. Radner (Eds.), *Decision and Organization* (pp. 161-176) Amsterdam: North-Holland.

Tune, G. S. (1964). A brief survey of variables that influence random generation. *Perception and Motor Skills, 18*, 705-710.

von Neumann, J., & Morgenstern, O. (1944). *Theory of Games and Economic Behavior.* Princeton University Press.

Wagenaar, W. A. (1972). Generation of random sequences by human subjects: A critical survey of the literature. *Psychological Bulletin, 77*, 65-72.

Ward, L. M. (1973). Use of markov-encoded sequential information in numerical signal detection. *Perception and Psychophysics, 14*, 337-342.

Ward, L. M., Livingston, J. W., & Li, J. (1988). On probabilistic categorization: The markovian observer. *Perception and Psychophysics, 43*, 125-136.

West, R. L. (1998a). Zero Sum Games as Distributed Cognitive Systems [Summary]. In *Proceedings of the Twentieh Annual Meeting of the Cognitive Science Society*. Mahwah, NJ: Erlbaum.

West, R. L. (1998b). Zero Sum Games as Distributed Cognitive Systems. In *Proceedings of the Complex Games Workshop*. Tsukuba, Japan: Electrotechnical Laboratory Machine Inference Group.

West, R. L. (1999). Simple Games as Dynamic Distributed Systems: A Neural Network Model of the Emergent Properties. Manuscript submitted for publication.