Cognitive Architectures, Game Playing, and Human Evolution

Content Areas: cognitive modeling, game theory, agents

Robert L. West,
Departments of Psychology & Cognitive Science, Carleton University,
robert_west@carleton.ca

Christian Lebiere,
Micro Analysis and Design, cl@cmu.edu

Dan J. Bothell,
Department of Psychology, Carnegie Mellon University, db30@andrew.cmu.edu

1 Introduction

Game playing is an excellent domain for researching interactive behaviors since any time the outcomes of the interactions between people are associated with payoffs the situation can be cast as a game. Because it is usually possible to use game theory (VonNeumann & Morgenstern, 1944) to calculate the optimal strategy, game theory has often been used as a framework for understanding game playing behavior in terms of optimal and sub optimal playing. That is, players who do not play according to the optimal game theory strategy are understood in terms of how they deviate from it. In this chapter we explore whether or not this is the right approach for understanding human game playing behavior, and present a different perspective, based on cognitive modeling.

Optimal, game theory models have been shown to be predictive of competitive strategies used by some animals (see Pool, 1995 for a review), leading to the argument that the process of evolution acts as a genetic algorithm for producing optimal or near optimal competitive behaviors. However, game theory models have not been very successful in predicting human behavior (Pool, 1995). In fact, psychological testing indicates that, from a game theory perspective, humans do not have the necessary cognitive skills to be good players. According to the classical game theory view, two abilities are needed to be a good game player (note, game theorists do not claim that game theory describes the cognitive process underlying game playing, however, these two abilities are necessary to play in the manner described by game theory): (1) the player needs the ability to calculate or learn the optimal probabilities for emitting each move, and (2) the player needs to be able to select moves at random, according to these probabilities. Humans are remarkably poor at both of these tasks. For example, in a simple guessing task in which a signal has an 80% chance of appearing in the top part of a computer screen and a 20% chance of appearing in the bottom, instead of adhering to the game theory solution and always guessing that the signal will be in the top part (for an optimal hit rate of 80%) people will fruitlessly try to predict when the signal will appear in the bottom part (for a hit rate of approximately 68%); which causes us to perform significantly worse than rats (Gazzaniga, 1998). Likewise, in addition to being poor at finding optimal probabilities, humans have been shown to be very poor at behaving randomly across a wide variety of tasks (see Tune, 1964, and Wagenaar, 1972 for reviews).

Given that humans are, arguably, the most successful species on earth, it does not seem reasonable that we should fail to fit the profile of a successful competitor. The answer to this problem lies in the unique adaptive strategy adopted by humans. In almost all cases, other creatures have evolved niche strategies. That is, they have adapted to compete as effectively as possible within particular environments and/or against particular opponents. These strategies tend to be near optimal, in the game theory sense, and also tend to be relatively inflexible. In contrast, humans have evolved to use learning, reasoning, problem solving, and creative thought to respond in highly adaptive ways across a wide variety of conditions.

From a game playing perspective, these two evolutionary strategies equate to two different types of players. As noted above, niche players can often be understood as optimal or near optimal players. Optimal players conform to game theory expectations in that (1) their choice of moves across time can be described in terms of selecting moves according to fixed probabilities and (2) these probabilities delineate an optimal or near optimal approach to the game. In contrast, the strategy of using some form of learning or thinking to try to improve the choice of future moves is a *maximizing* strategy. Maximal players do not use a fixed way of responding. Instead they attempt to adjust their responses to exploit perceived weaknesses in their opponent's way of playing. We argue that humans have evolved to be maximal rather than optimal players. That is, in competitive situations, humans attempt to exploit their opponent's weaknesses, rather than play optimally. Furthermore, we argue that evolution has evolved the human cognitive system to support a superior ability to operate as a maximizing player.

1.1 Maximal Versus Optimal

Maximal agents are potentially more effective than optimal agents against non-optimal agents. The optimal game theory solution is calculated by assuming that the opponent will play rationally. What this amounts to is an assumption that all players will assume that all other players will attempt to find the optimal strategy. If an opponent is using a sub optimal strategy the optimal player will generally fail to exploit it. For example, the game theory solution for the game of Paper, Rock, Scissors is to play randomly 1/3 paper, 1/3 rock, 1/3 scissors (in this game paper beats rock, rock beats scissors, and scissors beats paper). If an opponent plays 1/2 paper, 1/4 rock and 1/4 paper, the optimal strategy will tend to produce ties instead of the wins that could be produced by maximizing and playing scissors more. Nevertheless, it is also true that if a maximal agent plays against an optimal agent the best they can do is tie. However, keep in mind that for an optimal agent to be safe against all maximizing agents it needs the ability to behave truly randomly, something that may not be all that common in the natural world. Overall, we can characterize optimal agents as being designed to avoid loosing, while maximizing agents can be characterized as being designed to try to win by as much as possible, at the risk of losing.

1.2 Understanding Maximizing Strategies

Game theory provides a mathematical model for understanding and calculating optimal strategies. In this framework it is generally possible to calculate who should win, how often they will win, and how much they will win by. However, for games between maximizing players it can be very difficult to predict these things. The reason for this is that when two maximizing agents interact they form a dynamically coupled system. In order to adjust their behavior to exploit their opponent they have to sample their opponent's behavior to find a weakness. After they alter their behavior to exploit their opponent, the opponent will eventually detect the change (e.g., through sampling) and alter its behavior to exploit weaknesses in the new behavior. Thus, maximizing agents can end up chasing each other, trying to stay on top with the best strategy. This could result in an agent ending up in equilibrium, where the agent maintains a single strategy, or a limit cycle, where an agent repeatedly cycles through a limited set of strategies. However another possibility is that the coupled system, composed of the two interacting agents, could fail to settle into a stable pattern and instead produce a chaos-like situation (the term *chaos-like* is used instead of *chaos* since truly chaotic systems, i.e., systems that never repeat, exist only in mathematics or in physical, analogue systems. In this case, *chaos-like* is simply meant to refer to dynamic systems that appear to an observer to behave randomly).

Clark (1997, 1998) refers to these chaos-like interactions as reciprocal causation. Reciprocal causation is associated with emergent properties, that is, these systems often produce unexpected, higher level patterns of behavior. In terms of game playing, the ability of one player to beat another at a greater than chance rate is the higher-level pattern of interest. Clark (1997) also notes that, due to the chaos-like properties of reciprocal causation systems, it is often difficult to deliberately design systems to produce specific emergent properties. This is because predicting the results of these types of interactions is often mathematically intractable. To deal with this problem, maximizing strategies are usually studied by using computer simulations to create games between agents programmed with specific maximizing strategies.

This approach has been used by game theorists is to study the role of learning in game theory. A central question in this area of research has been whether or not players could learn the optimal move probabilities through their experience in a game. More specifically, if both players adjusted their move probabilities to create an advantage for themselves based on the history of their opponent's moves, would they eventually settle into an equilibrium equivalent to the game theory solution? If so, it would mean that the optimal game theory solution would still be relevant for understanding maximizers. However, research has shown that maximizers can co-evolve to non-optimal solutions (e.g., see Fudenberg and Levine, 1998; Sun and Qi, 2000), meaning that the optimal strategy is not predictive of behavior in these cases.

We also used the simulation approach, but with one important difference. Rather than adapting the basic game theory model to include learning, we based our model on psychological findings describing the way people process information in game-like situations. Thus we draw a distinction between *game theory maximizers* (i.e. the game theory model with the proviso that the move probabilities be learned) and *cognitive maximizers* (i.e., models based directly on the way human cognition works). Our contention is that these two approaches are very different and that the cognitive maximizer perspective is necessary for understanding human game playing behavior.

1.3 Experimental Psychology And Reciprocal Causation

Humans frequently interact in complex and dynamic ways. Despite this, experimental psychology is based almost exclusively on studying individuals in isolation, interacting with static situations (i.e., situations that do not feedback or do not feedback in a way that could produce reciprocal causation). This has allowed psychology to avoid the difficulties associated with studying complex dynamic systems, and to amass a large body of facts and models describing how people respond under these conditions. However, it may also be preventing psychology from forming a complete picture of human behavior. Hutchins (1995) has argued that much of what humans have achieved is due to distributed cognition rather than individual cognition - where distributed cognition refers to the fact that cognition (the processing of symbolic information) can occur across brains (linked by language and other means of communication). Likewise Clark (1997) has noted that much of human behavior seems to form reciprocal causation linkages to the world and to other humans (e.g., the economic system).

Others (e.g., Van Gelder & Port, 1995) have pointed to the limited number of studies showing that dynamic systems theory (i.e., mathematical, dynamic systems models) can be used to describe human behavior, and argued that traditional cognitive models (i.e., computational, symbolically based models) need to be abandoned in favor of dynamic systems models. We agree with Hutchins and Clark, that humans ultimately need to be understood in terms of the dynamic, interactive behaviors that make up most of their lives, but we disagree with the view that existing cognitive models need to be thrown out in favor of dynamic systems models. Instead we argue that experimental psychology has produced good models of specific cognitive mechanisms, and that these should form the building blocks for modeling complex interactive behavior.

However, interactive human behavior is often complex, involving more than one specific cognitive mechanism. Because of this need to go beyond the study of individual, isolated cognitive mechanisms, and the need to simulate interactions between agents, we argue that the use of cognitive architectures is the best way to proceed.

2 Cognitive Architectures

Cognitive architectures (specifically, production systems) were proposed by Newell (1973b) as a solution to the problems that he raised in a companion paper (Newell, 1973a) about the state of the study of cognition. The basic problem as he saw it was that the field of cognitive psychology practiced a strategy that was too much divide and too little conquer. Increasingly specialized fields were being carved out and esoteric distinctions being proposed, without any resolution that could lead to an integrated understanding of the nature of human cognition. While the extent to which our cognitive abilities result from specialized capacities or from general-purpose mechanisms remains a hotly debated question, Newell’s concept of cognitive architectures addresses the underlying systemic problem of unification by providing computational accounts of the findings of each specialized area in a comprehensive and integrated architecture of cognition. He later developed and proposed his own Soar architecture as a candidate for such a unified theory of cognition (Newell, 1990).

Cognitive architectures can provide some insights into the nature of cognition, but they do not constitute a panacea. Cognitive architectures specify, often in considerable computational detail, the mechanisms underlying cognition. However, performance in a given task depends not only on those mechanisms but also on how a given individual chooses to use them. Individual differences include not only fundamental capacities such as working memory or psychomotor speed, but also a bewildering array of different knowledge states and strategies. Limiting the complexity and degrees of freedom of such models is a major challenge in making cognitive modeling a predictive rather than merely explanatory endeavor.

Hybrid architectures (see Wermter & Sun, 2000, for a review) have become increasingly popular over the last decade to remedy the respective shortcomings of purely symbolic or connectionist approaches. Symbolic architectures (e.g. Soar) can produce very complex, structured behavior but find it difficult to emulate the adaptivity and robustness of human cognition. Connectionist approaches (e.g., see McClelland and Rumelhart, 1986) provide very flexible learning and generalization to new situations, but have not been successful in modeling complex, knowledge rich behavior.

2.1 ACT-R

ACT-R (Anderson & Lebiere, 1998) is a cognitive architecture developed over the last 30 years at Carnegie Mellon University. At a fine-grained scale it has accounted for hundreds of phenomena from the cognitive psychology and human factors literature. The most recent version, ACT-R 5.0, is a modular architecture composed of interacting modules for declarative memory, perceptual systems such as vision and audition, and motor systems, all synchronized through a central production system (see Figure 1). This modular view of cognition is a reflection both of functional constraints and of recent advances in neuroscience concerning the localization of brain functions.

ACT-R is a hybrid system that combines a tractable symbolic level that enables the easy specification of complex cognitive functions, with a subsymbolic level that tunes itself to the statistical structure of the environment to provide the graded characteristics of cognition such as adaptivity, robustness and stochasticity. The subsymbolic level is controlled by functions that control the access to the symbolic structures. As ACT-R gains experience in a task the parameter values of these functions are tuned to reflect a rational adaptation to the task (Anderson, 1990), where rational refers to a general ability to respond rationally to our environment, as opposed to a rational analysis of the specific task. Using this approach, Anderson (1990) demonstrated that characteristics of human cognition thought of as shortcomings could actually be viewed as optimally adapted to the environment. For example, forgetting provides a graceful implementation of the fact that the relevance of information decreases with time.

The symbolic level of ACT-R is primarily composed of *chunks* of information, and production rules that coordinate the flow of information and actions between modules based on the current goals of the system, also represented as chunks. Chunks are composed of a small number of pieces of information (typically less than half a dozen), which can themselves be chunks. Chunks stored in declarative memory can be retrieved according to their associated subsymbolic parameter called *activation*. The activation of a chunk is influenced by several factors that cause activation to increase with frequency of access, decay with time, and vary with the strengths of association to elements of the context and the degree of the match to requested patterns (chunks are requested by production rules). The chunk with the highest level of activation is the one that is retrieved.

Production rules are condition-action pairs that fire based on matching their *if* condition with chunks in the buffers providing the interface with the other modules. When production rules execute their *then* condition they change the information in these buffers. This act can trigger actions, request information, or change the current goal. Because several productions typically match in a cycle, but only one can fire at a time, a conflict resolution mechanism is required to decide which production is selected. Productions are evaluated based on their associated subsymbolic parameter called expected utility. The expected utility of a production is a function of its probability of success and cost (to accomplish the current goal). Over time, productions that tend to lead to success more often and/or at a lower cost, receive higher utility ratings. Both chunk activation and production utility include noise components so declarative memory retrieval and conflict resolution are stochastic processes (for a more extensive discussion on ACT-R see XXXXX in this book).

Methodology

In this chapter we want to show that humans are “good” maximal players, but there is no direct way to do this. As noted above, it is often not possible to calculate whether one maximizing strategy is better than another. Also, since different maximizing strategies may draw on different abilities, it is not possible, as it is with game theory, to identify the essential abilities and test them in isolation (in game theory these are: the ability to learn or calculate the right probabilities and the ability to play randomly). Our solution to this was to create a cognitive model of how people play games and then to play this model against artificial intelligence (AI) models designed to play a particular game as well as possible. Although providing qualitative rather than definitive answers, this approach has led to important insights in the area of *perfect information games*. Perfect information games are games where it is, in principle, possible to calculate the best move on every turn. One of the best-known examples is the game of *chess*, which has provided important insights into human cognitive abilities through the matches between humans and computers; another good example is the game of *go*. These games are too complex for even the fastest computer to come close to finding the best move for every situation, but it is possible for them to search very deeply into future possibilities. What surprised many was the enormous amount of computing power required to beat a skilled human. Even today it is debatable whether or not computers have truly surpassed the best humans in chess, and it is definitely not the case for go.

Game theory applies to *imperfect information games*. In imperfect information games it is not, in principle, possible to calculate the best move on every turn because that would require knowing what your opponent was going to do. For example, in Paper, Rock Scissors, if your opponent is going to play rock then your best move is to play paper, but you cannot be sure when they will play rock. Game theory is a way to calculate the optimal way to play for these types of games. Generally, it is assumed that people are poor at imperfect information games and can easily be beaten by a well-programmed computer. The main reason for this is probably that people are poor at the basic skills required to be an optimal player, while computers are ideal for optimal playing. Prior to having humans play against computers, similar assumptions were made about perfect information games because of the belief that perfect information games were all about how deeply a player could search a game tree (i.e., the outcome of future moves). Similarly, we believe that the current view of people as poor imperfect information players is based on an erroneous view of imperfect information games; specifically that game theory delineates the essential skills. Demonstrating that the way people play games competes well with AI models designed to play specific games would support our hypothesis. Alternatively, if we are wrong, the human model should be badly beaten by the AI models

4 How Do Humans Play?

The first question that we need to ask is, do people play games in the way described by game theory, since if they do we have no need for cognitive models. The standard game theory model requires that the players be able to select moves at random according to preset probabilities. However, research has repeatedly shown that people are very poor at doing this (see Tune, 1964, and Wagenaar, 1972 for reviews) suggesting that our evolutionary success is not based on this ability.

Instead of trying to learn advantageous move probabilities, people try to detect sequential dependencies in the opponent's play and use this to predict the opponent's moves (Lebiere & West, 1999; West & Lebiere, 2001). This is consistent with a large amount of psychological research showing that when sequential dependencies exist, people can detect and exploit them (e.g., Anderson, 1960; Estes, 1972; Restle, 1966; Rose & Vitz, 1966; Vitz & Todd, 1967). It also explains why people tend to do poorly on tasks that are truly random - because they persist in trying to predict the outcome even though it results in sub-optimal results (e.g., Gazzaniga, 1998; Ward, 1973; Ward, Livingston, & Li, 1988).

West and Lebiere (2001) examined this process using neural networks designed to detect sequential dependencies in the game of Paper, Rock, Scissors. The networks where very simple two layer networks rewarded by adding 1 and punished by subtracting 1 from the connection weights, which all started with a weight of 0. The inputs to the network were the opponent's moves at previous lags and the outputs were the moves the player would make on the current play (see Figure 2). West and Lebiere (2001) found four interesting results: (1) the interaction between two agents of this type produces chaos-like behavior, and this is the primary source of randomness; (2) the sequential dependencies that are produced by this process are temporary and short lived; (3) processing more lags creates an advantage; (4) treating ties as losses (i.e., punishing the network for ties) creates an advantage. West & Lebiere (2001) also tested people and found that they played similarly to a lag 2 network that is punished for ties. That is, people are able to predict their opponent's moves by using information from the previous two moves, and people treat ties as losses. Although both the network model and game theory predicted that people would play paper, rock and scissors with equal frequency, the network model predicted that people would be able to beat a lag 1 network that was punished for ties and a lag 2 network that was not punished for ties; while the game theory solution predicted they would tie with these opponents. The results showed that people were reliably able to beat these opponents, demonstrating that the game theory solution could not account for all the results.

4.1 The ACT-R Model

Although ACT-R was not designed to detect sequential dependencies, it turns out that there is a straightforward way to get the architecture to do this. The model learns sequential dependencies by observing the relationship between what happened and what came before on each trial. After each turn, a record of this is stored in the ACT-R declarative memory system as a *chunk*. Each time the same sequence of events is observed it strengthens the activation of that chunk in memory. Thus, chunk activation level reflects the past likelihood of a sequence occurring. For example, if the opponent's last move was P (where P = Paper, R = Rock, and S = Scissors) and the model was set to use information from the previous move (i.e., lag 1 information), then the model would choose one of the following chunks based on activation level: PR, PS, PP (where the first letter represents the opponent's lag 1 move and the second letter represents the expected next move). The model would then use the retrieved chunk to select its own move based on what it expected it's opponent to do. Thus if PR had the highest activation the model would play P to counter the expected move of R. The model would then see what the opponent actually did and store a record of it (e.g., assume the opponent played S, the model would then store PS), which would strengthen the activation of that sequence. Also, in addition to the correct chunks being strengthened on each trial, the activation levels of the chunks that are not used are lowered according to the ACT-T memory decay function (Figure 3 shows this process for a lag 2 model).

4.2 Accounting For Human Data

In theory, ACT-R represents fundamental cognitive abilities directly in the architecture, while learned abilities are represented as information processed by the architecture. The model described above is based directly on the ACT-R architecture and therefore represents a strong prediction about the way people detect sequential dependencies (i.e., because it is not influenced by assumptions about how learned information could influence the task). Also, it should be noted that our results do not depend on parameter tweaking. All parameters relevant for this model were set at the default values found to work in most other ACT-R models.

Simulations and testing with human subjects confirmed that the model could account for the human Paper, Rock, Scissors findings (Lebiere & West, 1999). This was very significant since the aspects of the architecture that we used were developed to model the human declarative memory system, not our ability to play games. It suggests that the evolutionary processes that shaped declarative memory may have been influenced by competition (in the game theory sense) for resources and mating privileges. It also indicates amazing design efficiency, as it suggests that humans use the same system for competition as they do for learning facts about the world.

The same model, without any changes other than adapting it to handle different games, has also been shown to account for batting results in baseball players (Lebiere, Gray, Salvucci, & West, 2003) and strategy shifts in 2X2 mixed strategy games, including the famous prisoner's dilemma (Lebiere, Wallach, & West, 2000). These findings indicate that this general mechanism is fundamental to human game playing abilities. However, we would not go so far as to claim that this simple mechanism could completely account for all human game playing. The structure of the ACT-R architecture itself suggests that under certain conditions people may learn specific production rules (using the procedural memory system) that can interact with or override the system we have described. Another possibility is that people may use the declarative memory system in different ways. For example, if a person does not have a strong feeling (activation strength) about the opponent's next move, they might instead opt to play a sequence that has caused the opponent to behave predictably in the past. Such sequences would also be learned through the declarative memory system. In game playing terms, having this type of flexibility is advantageous as it means that it would be difficult to develop systems that could routinely beat ACT-R models.

5 Comparison to other architectures

We chose ACT-R to model human game playing because of the substantial body of work showing that ACT-R is a good model of human cognition. However, it is not the case that ACT-R is the only architecture capable of playing in this way. Any architecture capable of detecting sequential dependencies could most likely be adjusted to produce similar results for individual games. In fact, as noted above, we have used both neural networks and ACT-R to model human playing. ACT-R is often contrasted with neural networks but the ACT-R declarative memory system possesses network-like abilities. The ACT-R model presented in this chapter can be thought of as roughly equivalent to a simple network (no hidden layer) with feedback that rewards the correct answer on each trial while the wrong answers are punished through the decay function. In addition to neural networks, hybrid architectures embodying some form of network (e.g., Clarion, see XXXX in this book) as well as models based directly on sequential dependency detection algorithms could potentially be adjusted to produce similar results (see Ward, Livingston, & Li, 1988 for an example of how this might be done with a sequential dependency detection algorithm). However, the ACT-R architecture can be viewed as a good choice for four reasons: (1) the architecture severely constrains how the declarative memory system could detect sequential dependencies, (2) it works with no parameter tweaking (all relative parameters were set to default values), (3) it locates the process within a well studied model of a particular brain function (4) the same process can also be used to explain other, non-game results, such as implicit learning (Lebiere & Wallach, 1998).

Models that do not play by detecting sequential dependencies may also be able to capture some game results. For example, the classic game theory model can capture the result that across time and across individuals, human players seem to play paper, rock and scissors with equal frequency. Also, ACT-R can be programmed to play through the production learning system rather than through the declarative memory system. The strategy shift in prisoners dilemma which can be fairly well accounted for using the ACT-R declarative memory system (Lebiere, Wallach, & West, 2000) can also be fairly well accounted for using the ACT-R production learning system (Cho & Schunn, 2002). Note that the production system model is the same general type as the maximizing game theory models mentioned above, where each move (represented by a production) has a certain probability of being chosen, and these probabilities are learned through experience. However, this approach does not account for the findings that humans use sequential dependency information and are bad at being random. Also, it is seems unlikely that this type of model could replicate the West & Lebiere (2001) data demonstrating that humans could beat some of the network models. This is because the only way to beat the network models was to somehow capitalize on the short lived sequential dependencies that they produced. However, it is possible that some people may play this way for some games. For example, some people may have well learned rules for cooperation that would influence how they play prisoner's dilemma, and would be more appropriately modeled through the ACT-R production system.

6 How Well Does ACT-R Play

We have argued, based on the evolutionary success of the human race, that the way people play games likely constitutes a good, general-purpose design for maximizing agents. To test this, we entered our ACT-R model in the 1999 International RoShamBo Programming Competition (RoShamBo is another term for Paper Rock Scissors). Although Paper, Rock, Scissors is a simple game, it is not easy to design effective maximizing agents for this game due to the reasons described above. The goal of the competition was to illustrate this fact and explore solutions (see Billings, 2000, for details and discussion).

Overall, ACT-R placed 13th out of 55 entries in the round robin competition (scores calculated based on margin of victory across games, e.g., +5 for winning by 5 and –5 for losing by 5). However, to get a better idea of how ACT-R compared to the other models we will focus on the open event, where ACT-R faced all the models. In this event ACT-R placed 15th in terms of margin of victory and 9th in terms of wins and losses. That is, the ACT-R model, with no modifications, was able to beat most of the other models.

To further test our claim we entered the same model in the 2000 International RoShamBo Programming Competition. However, the code for the winning program in 1999, which had been able to infer the ACT-R strategy well enough to beat it by a large margin, had been released (see Egnor, 2000). Therefore we expected a lot more programs would have this ability in 2000. To counteract this, we created a second model that retained the essential features of the first model but incorporated a strategy to prevent other programs from locking onto the ACT-R strategy. This model was called ACT-R-Plus. ACT-R-Plus simultaneously ran 30 ACT-R models that looked at both the opponent's history and its own history. The lags were set at 0, 1, 2, 3, 4, and 5 (lag = 0 would just keep track of what the most likely move is, regardless of history) and for each of these there was a version with noise on and noise off (the ACT-R chunk retrieval process involves a noise component that can be turned off). These were then combined with 3 strategies for choosing a move based on the prediction of the opponent's move: play the move that beats the move predicted, play the move predicted, or play the move that loses to the move predicted. As with the ACT-R model, the prediction with the highest activation value was chosen. Of course, ACT-R-Plus does not represent how humans play Paper Rock Scissors. Instead, it was an experiment in combining brute strength tactics with a human inspired architecture. In a sense, playing against ACT-R-Plus is like playing against a committee of agents, each with slightly different approaches as to how to use the ACT-R architecture to play the game.

In the round robin event, ACT-R came in 31st out of 64 while ACT-R-Plus came in 14th. In the open event ACT-R came in 32nd according to margin of victory and 28th according to wins and losses. ACT-R-Plus came in 9th according to margin of victory and 16th according to wins and losses. It was interesting to note that ACT-R was once again able to beat most of the models, despite the fact that code that could beat it had been released and had influenced many of the new models. However, since this program still placed 3^{d} in the competition, we speculate that in trying to improve on the code, many people actually made it worse. This again highlights the difficulties in designing maximizing agents.

The models in the competition could be divided into two types, *historical* models that searched for specific patterns in the history of the game, and *statistical* models that searched for statistical trends in the history of the game. To get a better idea of how well ACT-R performed, Figure 4 shows the open event results for ACT-R; ACT-R-Plus; the first placed model, which was historical; and the second place model, which was statistical. From this graph we can see that, although it was not able to exploit some models as well as the history model or the statistical model, ACT-R-Plus compares quite well. It mostly wins and when it loses it does not lose by much. ACT-R loses more but only the first placed history model is able to exploit it in a big way (this can be seen in the first point for ACT-R and the second big spike for the history model). Otherwise, overall, the performance of the basic ACT-R model is not bad, especially when you consider its relative simplicity and the fact that it was not designed for this competition.

7 Summary

When viewed from a traditional game theory perspective, humans do not appear to be particularly skillful game players. However, this is difficult to reconcile with our evolutionary success, which indicates that we are very effective competitors. We argued that this is because human game playing needs to be viewed as a maximizing strategy rather than the optimizing strategy suggested by traditional game theory analysis. However, it is difficult to evaluate the effectiveness of different types of maximizing strategies because competing maximizers can feed back on each other and form dynamically coupled systems which can give rise to emergent properties that are difficult to foresee (Clark, 1997). This was demonstrated in the results of the International RoShamBo Programming Competitions, which showed that even for the very simple game of Paper, Rock Scissors it is difficult to predict the results of this type of interaction.

In support of our position we reviewed a series of findings on human game playing abilities. Consistent with our view that humans are maximizing players we found that, under close examination, standard game theory models do not describe human game playing very well (at least for the games we investigated). Instead of trying to optimize move probabilities, humans try to maximize by exploiting the short lived sequential dependencies produced when they interact with another maximizing player (West & Lebiere, 2001). We also found that this type of interaction produces complex (chaos-like) behaviors and higher-level emergent properties resulting in one or the other player receiving an advantage. Following this we showed that these behaviors could be accounted for in a detailed and straightforward way by using the ACT-R cognitive architecture, and that the model could account for human behavior across a number of different games. This finding supports our contention that the human cognitive architecture, in addition to supporting individual activities, supports a level of functionality that can only be accessed by studying the dynamic interactions that occur between people. Finally, we demonstrated that the way humans play games, as represented by the ACT-R model, compares well to agents specifically created to play a particular game.

When considering the tournament results it is important to keep in mind that the ACT-R model was much simpler than the other models shown in Figure 4 and that the ACT-R model can play many different games without modifying the basic strategy. We also showed that the basic ACT-R model could be adapted to deal with specific limitations of the basic ACT-R model for a particular game (e.g., ACT-R –Plus). Although the adaptations that we made were not cognitively inspired, it is possible that with sufficient experience, humans could effectively augment their basic strategy. The main point however is that the general human strategy was competitive with and in many cases superior to AI strategies designed specifically for this game.

Finally, it is important to note that the same architectural components that we have shown to be important for game playing have also been shown to be important in a wide variety of other tasks unrelated to game playing (e.g., tasks involving problem solving and learning). Humans do not have a separate, dedicated system for game playing; we use the same cognitive system for a vast array of divergent tasks. Thus, the human cognitive system represents a highly efficient, multipurpose mechanism that has evolved to be as effective as possible across a wide variety of behaviors.

References

Anderson, J. R. (1990). *The Adaptive Character of Thought.* Hillsdale, NJ: Erlbaum.

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

Anderson, N. H. (1960). Effect of first-order probability in a two choice learning situation, *Journal of Experimental Psychology, 59*, 73-93.

Billings, D. (2000). Thoughts on RoShamBo. *The International Computer Games Association Journal, Vol. 23, No. 1*

Cho, K., & Schunn, C. D. (2002). Strategy shift in prisoner's dilemma through utility learning. Paper presented at the 9th Annual ACT R Workshop, Carnegie Mellon Univ., Pittsburgh, PA.

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.

Egnor, D. (2000). IOCAINE POWDER, *The International Computer Games Association Journal, Vol. 23, No. 1*

Estes, W. K. (1972). Research and theory on the learning of probabilities. *Journal of the American Statistical Association, 67,* 81-102.

Fudenburg, D., & Levine, D. K. (1998). *The theory of Learning in Games*. Cambridge, MA: The MIT Press.

Gazzaniga, M. S. (1998). The split brain revisited. *Scientific American,* July, 50-55.

Hutchins, E. (1995). *Cognition in the wild*. Cambridge, MA: The MIT Press.

Kelso, J. S. (1995). *Dynamic patterns: the self-organization of brain and behavior*. Cambridge MA: MIT Press.

Lebiere, C., Gray, R., Salvucci, D., & West, R. L. (2003). *Choice and Learning under Uncertainty: A Case Study in Baseball Batting*. Manuscript submitted for publication.

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.

Lebiere, C., Wallach, D., & West, R. L. (2000). A Memory-based Account of the Prisoner's Dilemma and Other 2x2 Games. *The proceedings of the Third International Conference on Cognitive Modeling*

Lebiere, C., & West, R. L. (1999). Using ACT-R to model the dynamic properties of simple games. *The Proceedings of Cognitive Science*.

McClelland, J. L. & Rumelhart, D. E. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* Cambridge, MA: Bradford Books.

Newell, A. (1973a). You can't play 20 questions with nature and win: Projective comments on the papers of this symposium. In W. G. Chase (Ed.), *Visual Information Processing* (pp. 283-310). New York: Academic Press.

Newell, A. (1973b). Production systems: Models of control structures. In W. G. Chase (Ed.), *Visual Information Processing* (pp. 463-526). New York: Academic Press.

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Cambridge University Press.

Pool, R. (1995). Putting game theory to the test. *Science*, *267*, 1591-1593.

Restle, F. (1966). Run structure and probability learning: Disproof of Restle's model. *Journal of Experimental Psychology, 72,* 382-389.

Rose, R. M., & Vitz, P. C. (1966). The role of runs of events in probability learning. *Journal of Experimental Psychology, 72,* 751-760.

Sun, R. & Qi, D. (2000). Rationality assumptions and optimality of co-learning *Proceedings of PRIMA'2000, Lecture Notes in Artificial Intelligence,* Heidelberg: Springer-Verlag.

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

Van Gelder, T., & Port, R. F. (1995). It's about time: An overview of the dynamic approach to cognition. In Port, R. F., & Van Gelder, T. (Eds.), *Mind as Motion* (pp 1-44). Mass: MIT.

Vitz, P. C., & Todd, T. C. (1967). A model of learning for simple repeating binary patterns. *Journal of Experimental Psychology, 75*, 108-117.

VonNeumann, J., & Morgenstern, O. (1944) *Theory of games and economic behaviour*. Princton, N. J.: Princton 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.

Wermter, S., & Sun, R. (2000). An overview of hybrid neural systems. In S. Wermter and R. Sun (Eds.): *Hybrid Neural Systems, Lecture Notes in Artificial Intelligence 1778,* Springer Verlag, Berlin.

West, R. L., & Lebiere, C. (2001). Simple games as dynamic, coupled systems: Randomness and other emergent properties. *Cognitive Systems Research, 1(4),* 221-239.

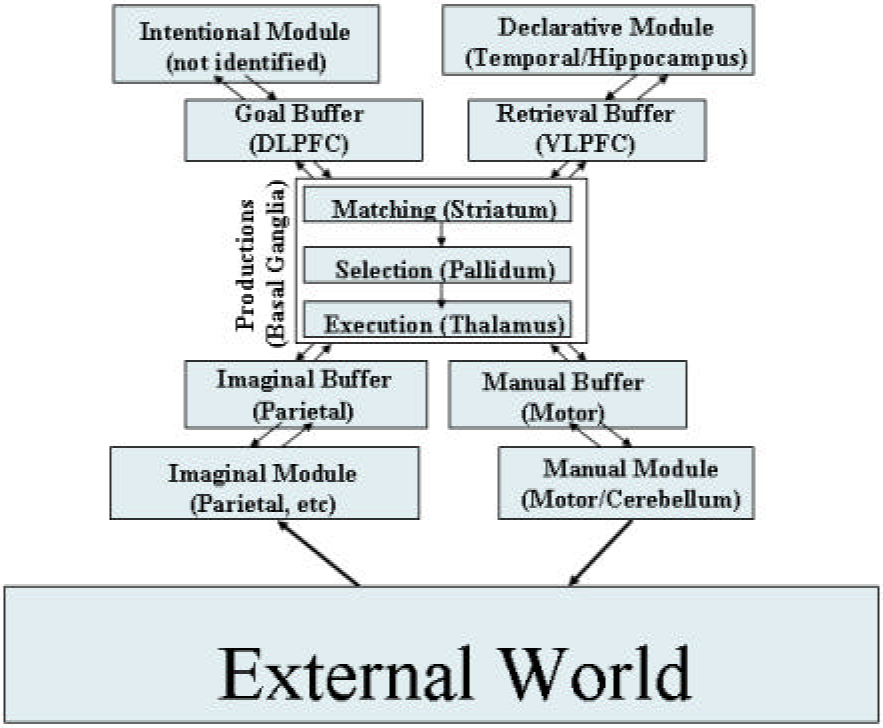

Figure 1. The component structure of ACT-R

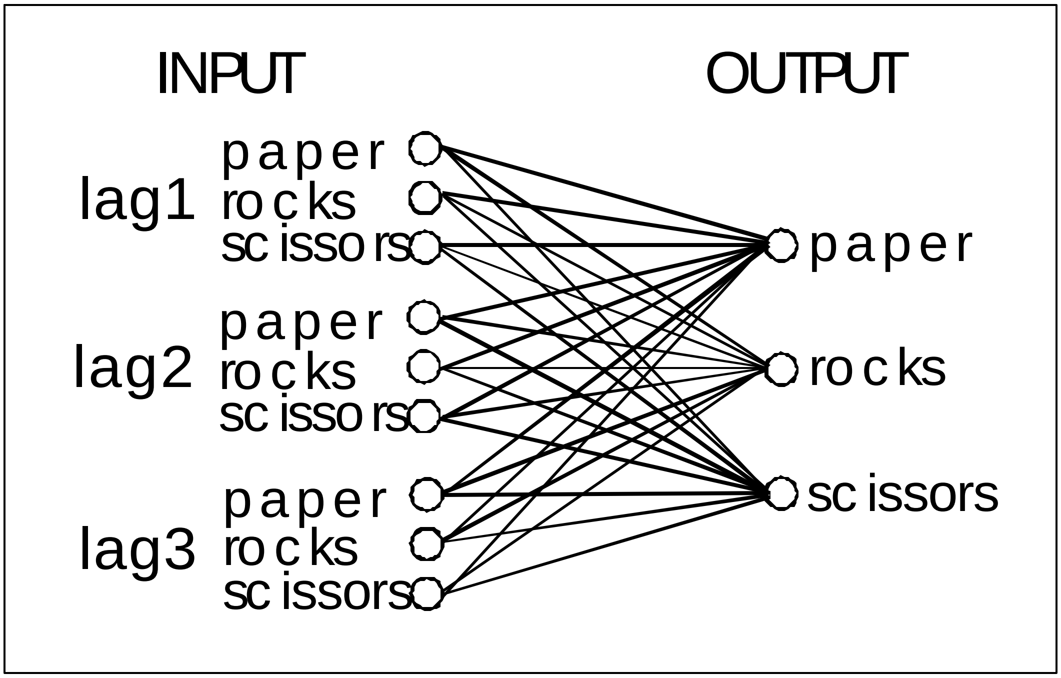

Figure 2. A lag 3 network model for playing paper, rock scissors. The model can be converted to a lag2 model by getting rid of the lag 3 inputs, or a lag 1 model by getting rid of the lag 2 and 3 inputs.

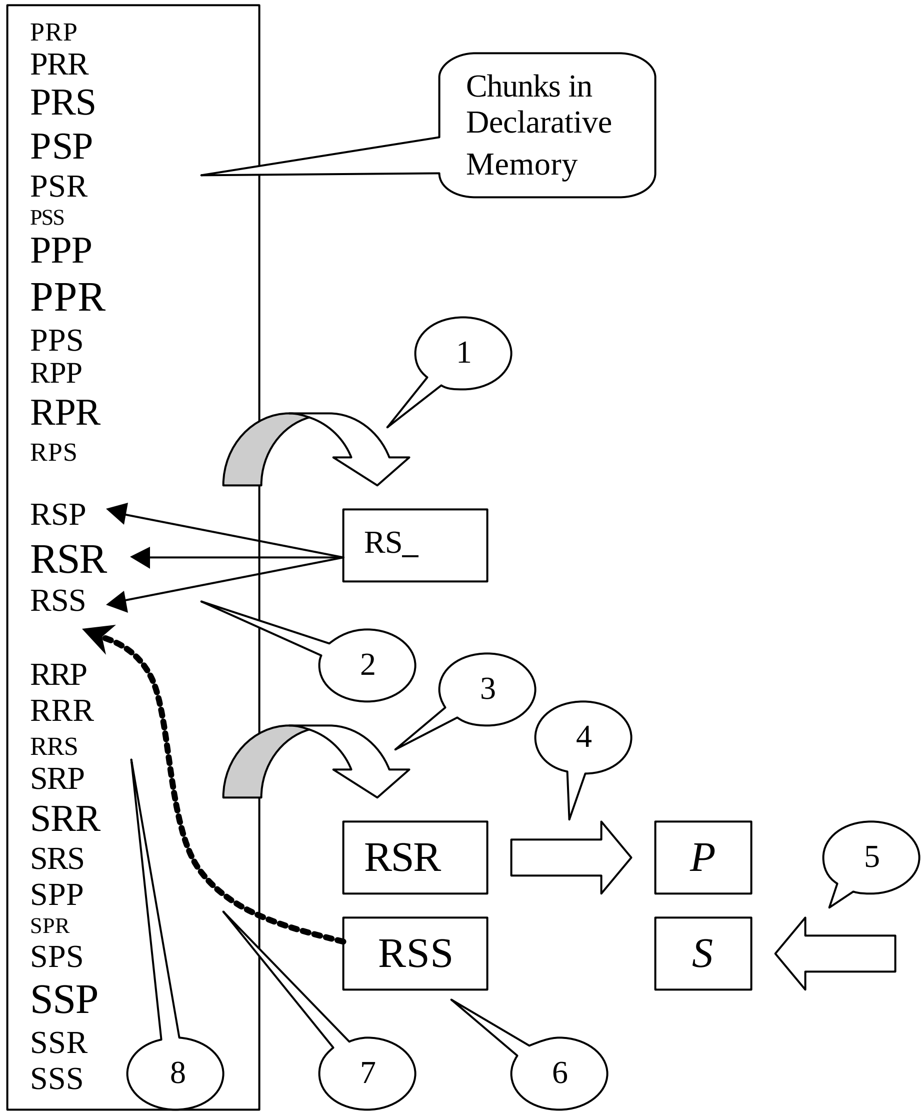

Figure 3. The process for an ACT-R, lag 2 model: (1) retrieve a chunk representing memory of the last two trials, with the chunk slot representing the current trial blank, (2) find the matching chunks, (3) retrieve the matching chunk with the highest activation level, (4) use the value in the current slot to predict the opponent's current move and play a move to counter it, (5) see what the opponent actually did, (6) create a chunk representing what actually happened, (7) put it into declarative memory where it will strengthen the activation of the chunk with the same slot values, (8) the activation level of all other chunks decays.

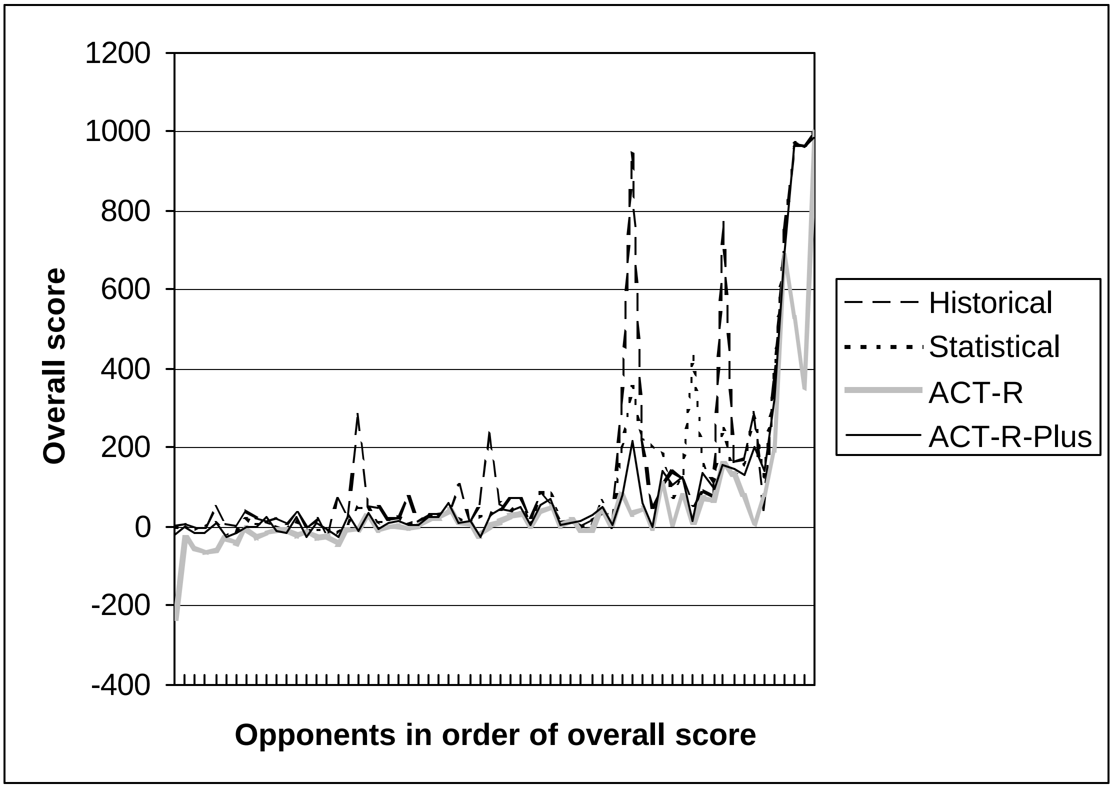

Figure 4. ACT-R results in the open event of the 2000 International RoShamBo Programming Competition