Game Playing: The Next Moves

Susan L. Epstein

Department of Computer Science
Hunter College and The Graduate School of The City University of New York
695 Park Avenue, New York, NY 10021
epstein@roz.hunter.cuny.edu

Abstract

Computer programs now play many board games as well or better than the most expert humans. Human players, however, learn, plan, allocate resources, and integrate multiple streams of knowledge. This paper highlights recent achievements in game playing, describes some cognitively-oriented work, and poses three related challenge problems for the AI community.

Game Playing as a Domain

Work on games has had several traditional justifications. Given unambiguous rules, playing a game to win is a well-defined problem. A game's rules create artificial world states whose granularity is explicit. There is an initial state, a state space with clear transitions, and a set of readily describable goal states. Without intervening instrumentation, games are also noise-free. For these reasons, as well as for their ability to amuse, games have often been referred to as "toy domains." To play the most difficult games well, however, a program must contend with fundamental issues in AI: knowledge representation, search, learning, and planning.

There are two principal reasons to continue to do research on games, despite Deep Blue's triumph (Hamilton and Hedberg 1997). First, human fascination with game playing is long-standing and pervasive. Anthropologists have catalogued popular games in almost every culture. Indeed, the same game, under various names, often appears on many continents (Bell 1969; Zaslavsky 1982). Games intrigue us because they address important cognitive functions. In particular, the games humans like to play are probably the ones we are good at, the ones that capitalize on our intellectual strengths and forgive our weaknesses. A program that plays many games well must simulate important cognitive skills. The second reason to continue game-playing research is that some difficult games remain to be won, games that people play very well but computers do not. These games clarify what our current approach lacks. They set challenges for us to meet, and they promise ample rewards.

This paper summarizes the role of search and knowledge in game playing, the state of the art, and recent relevant data on expert human game players. It then shows how cognitive skills can enhance a game-playing program, and poses three new challenge problems for the AI community. Although rooted in game playing, these challenges could enhance performance in many domains.

Search and Knowledge

In this paper, a *game* is a multi-agent, noise-free, discrete space with a finite set of objects (the *playing pieces*) and a finite, static set of rules for *play* (agents' serial behavior). The rules delineate where playing pieces can reside in the space, and when and how *contestants* (the agents) can *move* (transform one state into another). A *position* is a world state in a game; it specifies the location of the playing pieces and the agent permitted to act (the *mover*). The rules specify time limits for computation, a set of initial states, and a set of *terminal* states in which no agent is the mover. The rules assign to each terminal state a game-theoretic value, which can be thought of as a numerical score for each agent. The goal of each agent is to reach a terminal state that optimizes the game-theoretic value from its perspective. This definition includes finite-board games (e.g., tic-tac-toe and chess), games with imperfect information (e.g., poker and backgammon), and games with sequential team play (e.g., bridge), but excludes parallel activities (e.g., tennis and soccer).

A game may be represented by a *game tree*, in which each node represents a position and each link represents one action by one agent (called a *ply*). A *contest* at a game is a finite path through a game tree from an initial state. A contest ends at the first terminal state it reaches; it may also be terminated by the rules because a resource limit has been exceeded or because a position has repeatedly occurred. The *outcome* of a contest is the value of its terminal state, or the value (typically a draw) that the rules assign to a terminated contest.

An *optimal move* from position *p* is an action that creates a position with maximal value for the mover in *p*. In a terminal state, that value is determined by the rules; in a non-terminal state, it is the best result the mover can achieve if subsequent play to the end of the contest is always optimal. An *evaluation function* maps positions to scores for each agent. A *perfect* evaluation function preserves order among all positions' game-theoretic values; a *heuristic* one attempts to approximate them.

Given a subtree whose leaves are labeled by an evaluation function, a *minimax* algorithm backs those values up, one ply at a time, selecting the optimal move at each node

(Shannon 1950). With a small game tree, prior to any play one can minimax the values of all terminal nodes to compute the game-theoretic value of every node (*full retrograde analysis*) and cache those values with the optimal moves. The resultant table is a perfect evaluation function that can eliminate search during competition.

In a challenging game, a perfect evaluation function is unknown to human experts, and full retrograde analysis is intractable, as the search space sizes in Table 1 indicate. During play, *exhaustive search* (from the current position to only terminal states) followed by minimax could theoretically identify the correct move. The number of nodes visited during such search is dependent both on the game's *branch factor* (average number of legal moves from each position) and the depth of the subtree. Unless a contest is near completion or few pieces remain on the board, such a search is likely to be intractable.

Table 1: Estimated average branch factor and search space size for some challenging board games.

Game	*Board*	*Pieces*	*Branch factor*	*Space*
Checkers	8×8	32	8 – 20	$5 \cdot 10^{20}$
Chess	8×8	32	35	10^{120}
Shogi	9×9	40	80 - 150	10^{226}
Go	19×19	381	250	10^{360}

Resource conservation during game tree search has been attempted in both hardware and software. Hardware designed for a particular evaluation function can speed computation. Deep Blue's custom chess-searching chips, for example, enabled it to evaluate 50 to 100 billion moves in three minutes, sometimes to depths over 30 ply. Search algorithms can also improve efficiency. Some variations preserve exhaustive search's correctness: saving previously evaluated positions in a transposition table, the α-β algorithm (Slate and Atkin 1977), extensions along promising lines of play, and extensions that include forced moves (Anantharaman, Campbell, and Hsu 1990). Other search algorithms take conservative risks, pruning unpromising lines early (Berliner 1987) or seeking a stable heuristic evaluation (Beal 1990). Still others grow a best-first tree, guided by values estimated for the current leaves (Baum and Smith 1997; McAllester 1988; Palay 1985; Rivest 1987). Whatever its search mechanisms, however, a powerful game playing program typically plays only a single game, because it also relies on knowledge.

Knowledge can be incorporated into a game-playing program in three standard ways. First, formulaic behavior early in play (*openings*) is prerecorded in an *opening book*. Early in a contest, the program identifies the current opening and continues it. Second, knowledge about important principles in a game (e.g., control of the center) is embedded in a heuristic evaluation function. During play, the typical program searches to generate a subtree rooted at the current node, applies its heuristic evaluation function to the leaves of that subtree, and minimaxes those values to estimate the correct move. Finally, partial retrograde analysis may be performed offline, to calculate and store some true game-theoretic values and optimal moves. For nodes several ply from a terminal node, this is called a *closing book*. Because a heuristic evaluation function always returns any available closing book values, the larger the closing book, the more accurate the evaluation and the better a search engine is likely to perform. The best programs use both search and knowledge to win at difficult games.

The State of the Art

Checkers and chess

In 1994, Chinook became the world's champion checker player (Schaeffer 1997). Its opening book had 80,000 positions, and its 10-gigabyte closing book had some 443 billion positions, every position in which no more than eight checkers remain on the board. In the course of its development, it became clear that Chinook's ultimate prowess would be its knowledge base. As its closing book grew, Chinook improved. Eventually, with only the last 8 ply solved completely, Chinook defeated Marion Tinsley, long the human world champion.

In 1996, Garry Kasparov defeated Deep Blue, but in 1997 he lost to a program that used the same special-purpose hardware. In the intervening year, the program had received a substantial infusion of grandmaster-level knowledge. Its evaluation function had been strengthened, and its opening book had "a few thousand [items] chosen to match [its] style" plus statistics on grandmasters' opening play in 600,000 contests. Its closing book included all chess positions with five or fewer pieces, as well as number of moves to completion.

Chinook and Deep Blue are examples of *brute force*, the exploitation of search and knowledge on a heretofore-unprecedented scale. Each of them had a search engine that explored enormous subtrees, and supported that search with extensive opening and closing books. Each also had a carefully tuned, human-constructed, heuristic evaluation function, with features whose relative importance was well understood in the human expert community. There are, however, games whose large branch factors preclude deep search, games where human experts cannot provide all the knowledge computers need to win. Programs that play such games well have learned offline.

Backgammon, Othello, and Scrabble®

In 1997 Logistello defeated Takeshi Murakami, the human world Othello champion (Buro 1998), winning all 6 contests in the match. Logistello's heuristic evaluation function is primarily a weighted combination of simple patterns that appear on the board, such as horizontal or diagonal lines. (Parity and *stage*, how far a contest has progressed, are also included.) To produce this evaluation function, 1.5 million weights for conjunctions of these features were calculated with gradient descent during offline training, from analysis of 11 million positions. Although it uses a sophisticated search algorithm and a large opening book, Logistello's evaluation function is the key to its prowess. No more than 22 moves before the end of a contest, Logistello correctly computes the outcome.

TD-gammon, the best backgammon program by far, also relies on an evaluation function that was learned offline. TD-gammon narrowly lost at the AAAI-98 Hall of Champions to world champion Malcolm Davis by eight points over 100 contests. In backgammon, the dice introduce a branch factor of 400, rendering extensive, Deep-Blue-style search impossible. Instead, TD-gammon models decision making with a neural network pretrained by temporal difference learning on millions of offline contests between two copies of the program. During competition, TD-gammon uses its model to select a move after a 2-to-3-ply search. Since a reworking of its doubling algorithm, Tesauro estimates that TD-gammon has a slight advantage over the best human experts.

In the 1998 AAAI Hall of Champions, Maven defeated grandmaster Adam Logan 9 contests to 5 at Scrabble®, a game in which contestants place one-letter tiles into a crossword format. Maven is the best Scrabble® program, and among the top players of the game (Sheppard 1999). Scrabble® is subject to chance (tiles are chosen at random) and includes imperfect information (unplayed tiles are concealed). Maven uses a standard, game-specific move generator (Appel and Jacobson 1988) and the B* search algorithm (Berliner 1979); what distinguishes it from other programs is its learned evaluation function. Indeed, since their 1992 announcement, Maven's weights have become the standard for human and machine players. The current version also includes a probabilistic simulation of tile selection with 3-ply lookahead.

Although these three programs search and employ extensive knowledge, the changes that made them champions were ultimately changes to their learned evaluation functions. The creators of these programs gave learning a headstart: the right raw material from which to construct a powerful evaluation function. If the branch factor is too large, however, even a good evaluation function may be not be enough.

Current targets

Research on bridge and poker, card games of imperfect information, is ongoing. One program found five errors in the Official Encyclopedia of Bridge with a new heuristic technique (Frank, Basin, and Matsubara 1998). Loki now plays poker "better than the average club player" (Schaeffer 1999). At the same time, work on two perfect information games indicates that neither brute force nor offline learning will suffice.

Shogi is a chess-like game, with generals and lances instead of queens. In addition, most shogi pieces can be promoted to other pieces with more varied legal moves. Further complexity is introduced by the ability to *drop* (add to one's own forces any piece previously captured from the opposition) anywhere on the board. Shogi has a much larger branch factor than chess and, because of dropping, rarely ends in a draw. Positions are rarely *quiet* (having a relatively stable heuristic evaluation in a small subtree). Shogi has its own annual computer tournament, but no entry has yet played as well as a strong amateur.

Go presents an even greater challenge. Professional Go players develop slowly, typically after 10 years of full time study. Amateur Go players are ranked from 30 kyu to 1 kyu, and then from 1 dan to 6 dan. Professional players are ranked above that, from 1 to 9 dan. The Ing Cup, established in 1986, promises approximately $1.8 million for the first program to defeat Taiwan's three best 14-to-16 year-old players before 2000. Such a program would be ranked about 3 dan professional, a level hundreds of people attain. The best Go-playing program, however, now ranks only about 4 kyu. The Go-programming community generally acknowledges that the Ing Cup will expire without being won.

Heuristics are essential for games of incomplete information, which are NP-hard (Blair, Mutchler, and Liu 1993). They are also necessary to play shogi and Go well, but the requisite knowledge is inaccessible. Because their endgames are likely to have a larger branch factor and at least as many playing pieces, the generation of a useful closing book is intractable. Knowledge for a heuristic evaluation function is also problematic. In shogi, unlike chess, there is not even human consensus on the relative strength of the individual pieces (Beal and Smith 1998). In Go, the rules distinguish no *stone* (playing piece) from any other of the same color; a stone's significance is determined by the position. There are, moreover, thousands of possible Go features whose interactions are not well understood. To excel at their current targets, game playing programs will need something more.

Cognitive Science and Game Playing

Cognitive scientists seeking the source of human prowess have studied expert game players for more than a century. They take *protocols*, transcripts of peoples' verbal commentary during and after play. Cognitive scientists also use highly accurate devices to track experts' eye movements. Together, protocols and eye movement data suggest how experts play.

Perception and cognition

Neuroscientists have identified distinct regions in the human brain for visual perception and for high-level reasoning. In people, there is evidence that perceptual salience cues functional significance, thereby directing human attention (Tversky 1989). This is corroborated by timed images of a chess player's brain during decision making (Nichelli, Grafman, Pietrini, Alway, Carton, and Miletich 1994). Perception appears to begin first, followed somewhat later (and then in parallel with) reasoning and their integration. Indeed, protocols on game players regularly highlight *spatial cognition*, a process whereby people attend to visual features, determine their significance, and then value them accordingly. This is why people play better without blindfolds — vision not only focuses their attention, but is also an integral part of decision making.

Given the link between them, attention to perceptually salient features could cue learning about function (Tversky 1990). When a person needs to learn an evaluation function for playing the games considered here, she receives strong visual cues for the important features: the location of playing pieces and the shapes they form on the board. How might these cues be integrated with high-level reasoning?

A position can be described in terms of *patterns*, relations between playing pieces. When early results found no measurable difference in experts' and amateurs' search (de Groot 1965), Chase and Simon proposed that chess expertise lay in the acquisition and application of unordered spatial patterns they called *chunks*. Expert game players, they theorized, knew many important patterns, recognized them quickly, and associated a move selection mechanism with them. Better players would simply know more chunks, and propose moves appropriate to them**.**

Discrete chunks were not detected in Go (Reitman 1976), however, nor was pattern recognition alone able to account for differences detected more recently in the way better chess players search (Gobet 1998). Moreover, extensive studies indicate that expert Go players use not only static patterns but also dynamic ones (Yoshikawa and Saito 1997). With only one kind of piece for each contestant, Go appears to require a different kind of looking, one that dynamically invests stones with multiple roles, interleaving perception with cognition (Burmeister, Saito, Yoshikawa, and Wiles 1997). Professional Go players focused on lookahead and "good shape," an abstract image of certain parts of the current position, an image that lookahead manipulates and reformulates (Yoshikawa and Saito 1997). Go experts in those experiments looked at only a small part of the board, between stones, as if narrowing their options, and on recall often attributed a plausible meaning to a move. In response, several theories that annotate patterns have been proposed, including larger, frame-like structures called *templates* (Gobet and Simon 1996) and more flexible, more elaborate structures called *hybrid patterns* (Yoshikawa and Saito 1997). Whichever theory proves correct, vision is certain to play an integral role.

Less search and knowledge

Expert human game players have fast access to large stores of carefully organized, game-specific knowledge which they use both to focus their attention and to guide search. Compared to champion programs, however, human experts have far smaller opening and closing books. Their pattern-oriented representation for this data appears to be adapted to a specific game (Eisenstadt and Kareev 1975) and tailored only to positions that can arise during play (Chase and Simon 1973).

Compared to champion programs, human experts also consider fewer alternatives and search less deeply. Chess experts consider only a few moves, and rarely search more than 8 or 9 ply (Gobet 1998). Go experts perform relatively small searches, with average depth 4 and maximum branch factor 3 (Yoshikawa and Saito 1997). Human experts focus their attention on *candidate moves* (those worthy of search to evaluate them further) more quickly than novices. Eye movement studies confirm this in both chess and Go: experts look at very few points when they solve a test problem (de Groot 1965; Gobet 1998; Yoshikawa and Saito 1997). The Go players' speed is particularly remarkable: in 0.2—0.3 seconds the expert's eye fixates on the correct move. Accompanying protocols suggest that the selection of candidate moves involves both inference and pattern recognition.

Alternative cognitive mechanisms

There is increasing evidence, from a variety of domains, that people *satisfice*, that is, make a decision that is good enough. Satisficing often integrates a variety of strategies to accomplish problem solving (Biswas, Goldman, Fisher, Bhuva, and Glewwe 1995; Crowley and Siegler 1993; Ratterman and Epstein 1995). Satisficing may also include the ability to reason from incomplete or inconsistent information.

People develop expertise over time, through study and competition; they do not begin a new game as experts (Ericsson, Krampe, and Tesch-Römer 1993; Holding 1985). During development, people learn, particularly from their mistakes, whereas a deterministic program makes the same error repeatedly. Although TD-gammon, Logistello, and Maven have learned evaluation functions, they do not learn from their mistakes during competition, the way people would. Nor do people require millions of contests to achieve expert status. The initial learning curve in Go, for example, can be very rapid — a 5 kyu rank can be reached in several months (Saito and Yoshikawa 1997). In addition, as people have more experience with a task, their speed at it generally improves. In contrast, programs whose evaluation function functions rely upon more knowledge typically require more time at each search node, resulting in shallower search that can degrade performance (Anantharaman, Campbell, and Hsu 1990; Buro 1998).

Protocols also provide extensive evidence of high-level language concepts used to formulate subgoals, and of planning to reach those subgoals. Game playing experts summarize some of their knowledge this way. They use *tactics* (short-term plans) and *strategies* (long-term plans) to make decisions and to explain their behavior to each other. This is particularly accessible from protocols on masters playing Soudan Go, where each side is played in a separate room by a freely communicating team of two players (Yoshikawa, Kojima, and Saito 1998). The protocols demonstrate that "both sides clearly understand their opponent's intention and their understandings agree completely" (Saito and Yoshikawa 1997). Such evidence motivated the annotation of templates and hybrid patterns with variabilized plans and strategies.

Learning to Satisfice

Hoyle is a program whose cognitive orientation enables it to play a broad variety of games surprisingly well. Like people, but unlike the programs described thus far, Hoyle can play any two-person, perfect information, finite-board game, given the rules (Epstein, Gelfand, and Lock 1998). To date it has learned online, during competition, to play 18 games expertly in real time. The games, on two- and three-dimensional boards of various shapes, are drawn from anthropology texts, and are intended to capture problems that intrigue people. Although their game trees are relatively small (the largest has only several billion states), the speed with which Hoyle masters them, and its transparency, make the approach of interest here.

Hoyle begins any new game as a novice, but it knows what to learn. As it plays, it gradually acquires *useful knowledge*, probably correct and possibly applicable information from the contests it has experienced. Each kind of useful knowledge is prespecified, with its own representation and a procedure to learn it. An opening, for example, might be learned as a tree by rote.

Hoyle also begins with a set of move-selection rationales called *Advisors.* Given the acquired useful knowledge for a game and a position from it, an Advisor can recommend or caution against any number of moves by generating *comments*. Advisors are subject to resource limits, but not to any uniform decision process or representation.

Advisors are categorized into three tiers. Those in tier 1 rely on shallow search and learned game-theoretic values to provide perfect guidance on a single move, mandating its selection or avoidance. Tier-1 Advisors make easy decisions quickly, and prevent obvious blunders. Victory, for example, makes an immediately winning move. All other Advisors are heuristic, either because they depend upon induced useful knowledge or because their reasoning method is approximately correct. Material, for example, encourages piece capture in tier 3. Tier-2 Advisors advocate a sequence of moves (a *plan*), while tier-3 Advisors advocate a single move.

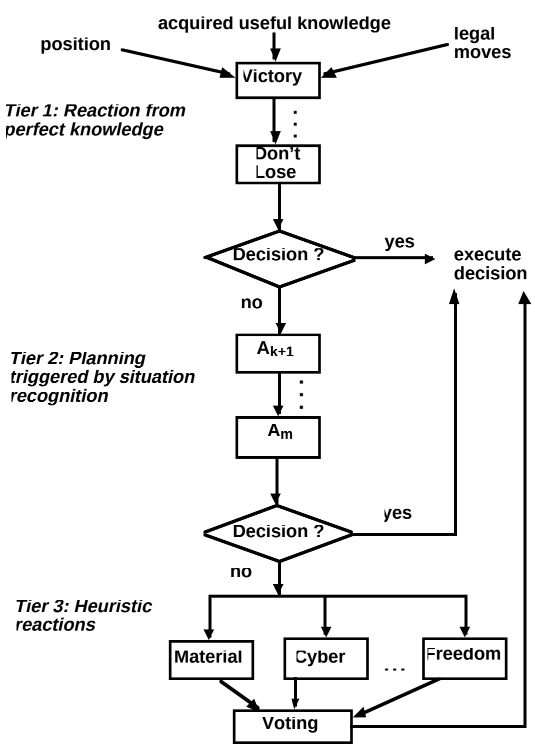

Figure 1: A schematic for Hoyle's move selection.

Comments are combined to select a move, as shown in Figure 1. Advisors are consulted serially in tiers 1 and 2, where the first Advisor able to determine the next move does so. In tier 3 Advisors are consulted in parallel, and a weighted combination of their comment strengths selects the best move. Weights are both game-dependent and stage-dependent, and are learned.

On a few simple games, Hoyle has been cognitively validated, that is, its learning and decision making have been shown similar to that of people (Ratterman and Epstein 1995). Indeed, the program intentionally has many of the hallmarks of human expert play:

- Hoyle satisfices. While it decides in real time, it tolerates incomplete and incorrect information, entertains a variety of conflicting rationales simultaneously, and degrades gracefully in unfamiliar situations.
- Hoyle learns quickly. It usually achieves its expertise in less than 100 contests.
- Hoyle is transparent. It can explain its moves through a natural language interpretation of its Advisors' comments.
- Hoyle integrates visual perception with high-level reasoning. Its allowance for multiple representations, in useful knowledge and in Advisors, supports this. The current position, for example, can be simultaneously represented as a list, an array, a bipartite graph, a set of fixed-shape patterns, and a set of piece-delineated territories.
- Hoyle links perception with function. It learns visually-based, high-level, game-dependent concepts as game-specific Advisors for tier 3. For example, it can learn that the consolidation of territory is advisable in a game, determine how that territory should be demarcated, and create and use an Advisor to encourage that behavior.
- Hoyle plans. It acquires both specific and tactical move sequences as tier-2 Advisors.
- Hoyle uses high-level concepts. Some, such as blocking a winning move, are game-independent, and prespecified in tier 1. Other, game-dependent ones are learned from perceptual cues, as described above.

New Game Playing Challenges

Results with Hoyle suggest ways in which a program might be more like a human champion. Hoyle's penchant for useful knowledge and multiple heuristic opinions is likely to prove helpful in addressing the following challenge problems. Each increases with difficulty, of course, according to the target game.

Problem 1: Model a contestant

Strategic and tactical decisions are often affected by the nature of one's problem solving environment. An important factor in game playing decisions, for example, should be the nature of one's opponent. The first challenge problem is to construct a model of a particular contestant in a two-person game. Given a suite of contests played by a single person or program against various competitors, the program should produce natural language that characterizes the opponent's current play in a contest (e.g., "Black isn't interested in cashing in his initiative with 19...Bxd4? 20 cxd4 Qxd4 when White has excellent play for the pawn"). It should also identify general strengths, weaknesses, and predilections, such as "In the opening, Kasparov plays black like white," "Karpov strives to improve his position little by little, maneuvering subtly, making his threats," "He is *always* attacking, often from directions his opponents hadn't considered, playing moves that have subtle multiple threats," and "You can tell from [his opponents'] moves they are scared. Their attacks are wild and hopeless or else very timid." (Waitzkin 1990).

Some groundbreaking work on this has been done in poker (Billings, Papp, Schaeffer, and Szafron 1998) and Go (Isozaki and Katsuno 1996). The generation and analysis of summary statistics would be a reasonable way to continue. Plan detection is required, as well as some representation for volition, aggression, and risk. Knowledge about emotions and the ability to generate metaphors should prove helpful.

Problem 2: Annotate a contest

People annotate a chess contest both move by move (e.g., "Far from losing a tempo, Black has really gained time since the knight stood better on d4 than b3") and with tactics and strategy (e.g., "try to play d2-d4, bring the knight on b1 to e2, and follow up with Nf3-e5, playing on the dark squares.") Annotations contain commentary and often diagrams. Since 1993, the International Computer Chess Association has awarded an annual prize to the program that best annotates an input chess contest in real time, but entries are few and quite machine-like (Bjornsson and Marsland 1998). Programs are encouraged to propose variations, include written comments, and provide explanatory diagrams.

The second challenge problem is to annotate a contest so well that people cannot distinguish it from a human-generated, popularly published annotation, say from *The New York Times.* A solution must generate both natural language and appropriate diagrams. It will also require high-level concepts appropriate to the game, context-sensitive perspectives, and the induction of a model for each contestant as an agent with intentions and the ability to deceive.

Problem 3: Teach a game

One measure of human expertise is the ability to convey knowledge. The third challenge problem is to have an expert game playing program (the *instructor*) teach its skill to a person or another program (the *student*). To do this well, the instructor must first analyze and represent its own knowledge. Next, the instructor would model the student's knowledge, diagnose its weaknesses, and formulate a curriculum. Instruction could be presented as positions, lines of play, or entire contests, all couched in appropriate natural language. An analysis of the student's learning style is likely to prove helpful. Once the student consistently plays as well as the teacher, the problem will be solved.

Conclusion

Champion programs thus far lack cognitive features often cited as hallmarks of human intelligence: online learning, planning, and the ability to communicate knowledge. There is a growing interest, however, in reactive, hierarchical satisficers like Hoyle, and in the kind of agent interaction and modeling targeted by the challenge problems posed here (Dobson and Forbus 1999). We can look forward, therefore, to programs that narrow their options, learn from their mistakes, model other agents, explain their rationales, and teach us to play better.

Acknowledgments

This work was supported in part by NSF grant #9423085. Thanks to Michael Buro, Murray Campbell, Takuya Kojima, Jonathan Schaeffer, Brian Sheppard, Gerry Tesauro, and Atsushi Yoshikawa for generously sharing their substantial expertise. Discussions with Hans Berliner, Ian Frank, Jack Gelfand, Esther Lock, and Donald Michie have inspired and informed much of this work.

References

Anantharaman, T., Campbell, M. S. and Hsu, F.-h. 1990. Singular Extensions: Adding Selectivity to Brute-Force Searching. *Artificial Intelligence*, 43(1): 99-110.

Appel, A. W. and Jacobson, G. J. 1988. The World's Fastest Scrabble Program. *Communications of the ACM*, 31(5): 572-578.

Baum, E. B. and Smith, W. D. 1997. A Bayesian Approach to Relevance in Game Playing. *Artificial Intelligence*, 97: 195-242.

Beal, D. and Smith, M. 1998. First results from Using Temporal Difference learning in Shogi. In *Proceedings of the First International Conference on Computers and Games*. Tsukuba, Japan.

Beal, D. F. 1990. A Generalised Quiescence Search Algorithm. *Artificial Intelligence*, 43(1): 85-98.

Bell, R. C. 1969. *Board and Table Games from Many Civilizations*. London: Oxford University Press.

Berliner, H. J. 1979. The B*Tree Search Algorithm: A Best-First Proof Procedure. *Artificial Intelligence*, 12(1): 23-40.

Berliner, H. J. 1987. Some Innovations Introduced by

HITECH. *ICAA Journal*, 18(2): 71-76.
Billings, D., Papp, D., Schaeffer, J. and Szafron, D. 1998. Opponent Modeling in Poker. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence,* 493-499. Madison: AAAI Press.
Biswas, G., Goldman, S., Fisher, D., Bhuva, B. and Glewwe, G. 1995. Assessing Design Activity in Complex CMOS Circuit Design. In P. Nichols, S. Chipman, & R. Brennan (Ed.), *Cognitively Diagnostic Assessment,* Hillsdale, NJ: Lawrence Erlbaum.
Bjornsson, Y. and Marsland, T. 1998. Fritz 5.0 Wins the 1997 Herschberg Best-Annotation Award. *ICCA Journal*, 20(1): 65-66.
Blair, J., Mutchler, D. and Liu, C. 1993. Games with Imperfect Information. In *Proceedings of the Fall Symposium on Games: Planning and Learning,* 59-67.
Burmeister, J., Saito, Y., Yoshikawa, A. and Wiles, J. 1997. Memory Performance of Master Go Players. In *Proceedings of the IJCAI Workshop on Using Games as an Experimental Testbed for AI Research*.
Buro, M. 1998. From Simple Features to Sophisticated Evaluation Functions. In *Proceedings of the 1st International Conference on Computers and Games*. Tsukuba.
Chase, W. G. and Simon, H. A. 1973. The Mind's Eye in Chess. In W. G. Chase (Ed.), *Visual Information Processing*, 215-281. New York: Academic Press.
Crowley, K. and Siegler, R. S. 1993. Flexible Strategy Use in Young Children's Tic-Tac-Toe. *Cognitive Science*, 17(4): 531-561.
de Groot, A. 1965. *Thought and Choice in Chess.* The Hague: Mouton.
Dobson, D. and Forbus, K. (1999). *Proceedings of the AAAI 1999 Spring Symposium on Artificial Intelligence and Computer Games*.
Eisenstadt, M. and Kareev, Y. 1975. Aspects of Human Problem Solving: The Use of Internal Representations. In D. A. Norman, & D. E. Rumelhart (Ed.), *Explorations in Cognition,* 308-346. San Francisco: Freeman.
Epstein, S. L., Gelfand, J. and Lock, E. T. 1998. Learning Game-Specific Spatially-Oriented Heuristics. *Constraints,* 3(2-3): 239-253.
Ericsson, K. A., Krampe, R. T. and Tesch-Römer, C. 1993. The Role of Deliberate Practice in the Acquisition of Expert Performance. *Psychological Review*, 100(3): 363-406.
Frank, I., Basin, D. and Matsubara, H. 1998. Finding Optimal Strategies for Incomplete Information Games. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence,* 500-507. Madison: AAAI Press.
Gobet, F. 1998. Chess Players' Thinking Revisited. *Swiss Journal of Psychology*, 57(1): 18-32.
Gobet, F. and Simon, H. 1996. Templates in Chess memory: A Mechanism for Recalling Several Boards. *Cognitive Psychology,* 31: 1-40.
Hamilton, C. M. and Hedberg, S. 1997. Modern Masters of an Ancient Game. *AI Magazine,* 18(4): 11-12.
Holding, D. 1985. *The Psychology of Chess Skill*. Hillsdale, NJ: Lawrence Erlbaum.
Isozaki, H. and Katsuno, J. 1996. A Semantic Characterization of an Algorithm for Estimating Others' Beliefs from Observation. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence,* 543-549. MIT Press.
McAllester, D. A. 1988. Conspiracy Numbers for Min-Max Search. *Artificial Intelligence*, 35(3): 287-310.
Nichelli, P., Grafman, J., Pietrini, P., Alway, D., Carton, J. and Miletich, R. 1994. Brain Activity in Chess Playing. *Nature*, 369: 191.
Palay, A. J. 1985. *Searching with Probabilities.* Pitman.
Ratterman, M. J. and Epstein, S. L. 1995. Skilled like a Person: A Comparison of Human and Computer Game Playing. In *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society,* 709-714. Pittsburgh: Lawrence Erlbaum Associates.
Reitman, J. S. 1976. Skilled Perception in Go: Deducing Memory Structures from Inter-Response Times. *Cognitive Psychology*, 8: 36-356.
Rivest, R. L. 1987. Game Tree Searching by Min/Max Approximation. *Artificial Intelligence*, 34(1): 77-96.
Saito, Y. and Yoshikawa, A. 1997. Go as a Testbed for Cognitive Science Studies. In *Proceedings of the IJCAI-97 Workshop on Using Games as an Experimental Testbed for AI Research*.
Schaeffer, J. 1997. *One Jump Ahead: Challenging Human Supremacy in Checkers*. New York: Springer-Verlag.
Schaeffer, J. (1999). Personal communication.
Shannon, C. E. 1950. Programming a Computer for Playing Chess. 41(256): 275.
Sheppard, B. (1999). Personal communication.
Slate, D. J. and Atkin, L. R. 1977. CHESS 4.5 - The Northwestern University Chess Program. In P. Frey (Ed.), *Chess Skill in Man and Machine,* Berlin: Springer.
Tesauro, G. 1995. Temporal Difference Learning and TD-Gammon. *CACM,* 38(3): 58-68.
Tversky, B. 1989. Parts, Partonomies, and Taxonomies. *Developmental Psychology*, 25(6): 983-995.
Tversky, B. 1990. Where Partonomies and Taxonomies Meet. In S. L. Tsohatzidis (Ed.), *Meanings and Prototypes - Studies in Linguistic Categorization,* 334-344. London: Routledge.
Waitzkin, F. (1990). *Kasparov*. The New York Times Magazine: 28-30, 60-64, 86-87.
Yoshikawa, A., Kojima, T. and Saito, Y. 1998. Relation between the Way of Using Terms and the Skill - from Analysis of Protocols of the Game of Go. In *Proceedings of the First International Conference on Computers and Games*, 211-227. Tsukuba.
Yoshikawa, A. and Saito, Y. 1997. Hybrid Pattern Knowledge: Go Players' Knowledge Representation for Solving Tsume-Go Problems. In *Proceedings of the South Korean International Conference on Cognitive Science*.
Zaslavsky, C. 1982. *Tic Tac Toe and Other Three-in-a-Row Games, from Ancient Egypt to the Modern Computer*. New York: Crowell.

FOR SLIDES

There is evidence that Go players shift their attention to various parts of the Go board (furikawari=neighboring shift)

in some manner less opaque to humans, although some can be extracted by “reading” TD-gammon’s weights (Tesauro 1995)

Fortuitously (or perhaps of necessity) there is in the games we play a strong relationship between physical proximity and function (with the exception of knight-like pieces).

There are even terms in Go for crucial stones and unimportant ones.

The efficient identification of territory Mueller

No orientation of go board and some peripheral stones may be ignorable

For slides: “The real art in chess is to evaluate the factors because they are so different. What is more important, one pawn or the open line? What’s more important, the weak position of your king or some initiative on the queenside?” “At the highest level, chess is a talent to control unrelated things.” “Some positions are so complex that you cannot calculate even two moves ahead.”

Karpov continued: He avoids weaknesses and waits for his opponent to make mistakes. He works to neutralize threats and gain control of certain squares and files, so that with his superior technique he will be able to win in the end game.

Re modeling cite Isozaki cite

• “… a ferocious offensive player, a bold risk-taker who will sacrifice pieces for positional advantages. He is *always* attacking, often from directions his opponents hadn’t considered, playing moves that have subtle multiple threats. Thus he prefers open games, where the middle of the board is unclogged, pieces can move freely, the positions are double-edged and the possibilities are vast…”

Its safe to say that the programs outplay the people, and that the programs don’t play perfectly. (Buro’s email)