From: AAAI Technical Report WS-97-04. Compilation copyright © 1997, AAAI (www.aaai.org). All rights reserved.

The Future of Chess-Playing Technologies and the Significance of Kasparov Versus Deep Blue

Dennis DeCoste
Jet Propulsion Laboratory / California Institute of Technology
4800 Oak Grove Drive; Pasadena, CA 91009
http://www-aig.jpl.nasa.gov/home/decoste/, decoste@aig.jpl.nasa.gov

Abstract

In this paper we argue that the recent Garry Kasparov vs. Deep Blue matches are significant for the field of artificial intelligence in several ways, including providing an example of valuable baseline benchmarks for more complex alternatives to contrast and justify themselves. We will also briefly summarize some of the latest developments on computer chess research and highlight how our own work on a program called Chester tries to build on those developments to provide such justifications.

Introduction

Since "falling from grace" (DS90) by the late 1980's, computer chess has recently received considerable attention in the popular media due to the 1996 match between IBM's Deep Blue chess machine and Garry Kasparov, the reigning World Champion. Furthermore, researchers of artificial intelligence seem to be increasingly willing to cite Deep Blue as an example of success (e.g. (SBD+96)). The topic of computer chess, and game-playing in general, has also enjoyed a considerable resurgence of significant and creative research, particularly in the areas of search and machine learning (as we shall discuss later).

In our view this reemergence of computer chess was somewhat simply a matter of time — requiring both maturity of technologies and better appreciation by AI researchers of the importance of empirical evaluation of comparative performance on problems with well-defined success criteria.

In this paper, we will argue that the Garry Kasparov versus Deep Blue match ("GK vs DB") has great significance to AI in several ways. We will directly avoid philosophical questions such as whether DB is a "thinking machine" or how we might test such a proposition. However, we will argue that the development of Deep Blue represents a legitimate, and in some ways even a role model, example of the future of AI research. We will also summarize some recent developments in relevant AI research areas, including our own.

Significance for AI

Appreciating the significance of GK vs DB for AI requires assessing the significance of four distinct items:

1. the task of computer chess,
2. the solution instance represented by Deep Blue,
3. the matches between GK vs DB per se,
4. the closed-nature of Deep Blue development.

We explore these issues in turn below.

Significance of Computer Chess

Criticisms of computer chess that lead to its fall from grace partly arose from growing pressure to fund and conduct research that directly addressed "real world" problems. Much early work was particularly vunerable to such attack, due to the predominance of manual crafting and tuning of search and evaluation functions that seemed required for competitive performance. Recent advances in software engineering, search, and machine learning technology promise to make these issues less germane in the future.

More fundamental has been the failure to address the issue of scaling-up. As Schank forcably argued in (Sch91) a key distinction between AI and engineering is that AI strives foremost for representations and algorithms that scale-up gracefully with task complexity. Recent success of predominately brute-force approaches such as Deep Blue seriously bring into question whether chess is complex enough to require more general AI techniques which scale-up. Indeed, many researchers on game-playing have shifted to games such as Go, whose sheer complexity seems certain to protect it from brute-force approaches. Others have argued for artificial generalizations, such as META-GAME (Pel94), which would better justify the use of learning and sophisticated search.

Some argue that chess is simply not a rich and important enough problem to warrant significant attention. However, the visibility and attention of the GK vs DB match demonstrates that chess performance remains a valued and easy-to-grasp metric. Furthermore, recent work on "rational search" (Bau92) highlights the fact that even though chess is a deterministic game, the importance of reasoning with uncertainty, which dominates many real-world tasks, arises even in chess.

Significance of Deep Blue

To evaluate the extent to which Deep Blue and other computer chess work is "AI", it is useful to recall that AI has several components, most notably: knowledge representation schemes, search, and learning. The best performing computer chess projects have often required significant manual knowledge encoding/tuning per se, causing some doubt of their being AI.

In contrast, the brute-force approach of Deep Blue clearly focusses on the issue of search over knowledge. However, Deep Blue's contributions to our understanding of this issue are debatable. For example, recent work has shown that there appear to be significant diminishing returns in chess around ply 10 (JSB^+), much as that occuring at shallower plys for simpler games such as checkers. Thus, it may be the case that continued work along Deep Blue's brute force lines will offer very little additional insight. Indeed, the designers of Deep Blue have publically stated that for the upcoming May 1997 rematch, the enhancements to Deep Blue have focussed more on improved knowledge than search per se. Whether such knowledge engineering will be of the unscalable and chess-specific sorts that made early chess work easier to dismiss or not remains to be seen.

Significance of GK vs DB

In our view, the largest significance of Deep Blue is in fact in the GK vs DB matches per se. The role and utility of "competitions" in driving and evaluating AI research seems to be getting increased acceptance in recent years, in part due to the easier disemination of data sets and domain knowledge via the advent of the World Wide Web. For example, the Sante-Fe Institute recently organized a successful competition to compare techniques on blind time-series prediction tasks over several data sets (WG94).

Of course, the history of computer chess is full of competitions, most notably the ACM tournaments. However, competition among relatively weak computer programs, as should be expected, tends to overly-reward short-term competitive advantages, such as cooking opening books. Such tricks are of little use against a flexible opponent such as Kasparov, particularly given that he is well-versed in the weaknesses and tendencies of existing computer chess programs.

It was thus of considerable interest when Kasparov proclaimed in his TIME article after the 1996 match that "I could feel — I could smell — a new kind of intelligence across the table" (Kas96). His surprising initial loss in Game 1, due to under-appreciating the effectiveness of Deep Blue's search, and his final crushing victory due to vastly superior positional play in Game 6 were dramatic contrasts in the current strengths and weaknesses of human and computer approaches.

We believe that similar evaluations will play an increasingly common role in future development of AI. First, because it is a natural human curiousity to want to compare machine and human approaches to achieving intelligent behavior. Second, because such comparisons can in fact provide useful scientific feedback. Similar feedback during the development of medical expert systems, for example, seems to have been a significant influence in the development of approaches to reasoning under uncertainty, such as Bayesian belief networks, that are now popular and promising in AI research.

Also, the importance of visibility can perhaps not be underestimated. Indeed, much of the public before the 1996 GK vs DB match seemed to have believed that computers had already "solved chess", or at least were better than all humans. Anyone who wants assurance that AI is still no match for human intelligence and flexibility need look no further than Game 6.

In short, we view the current significance of Deep Blue to be largely in providing some highly visible data points to better understand how far largely brute-force search can go in one historically-significant and complex domain. We believe it extremely important that such work continue to parallel that of more fundamental AI research. At the very core of the notion of intelligence is the notion of speed of computation. Indeed, typical IQ tests do not allow one to bring the test home and work it out in leisure over the course of a year. In this sense, better case studies into when specific styles of search and shortcuts are sufficient for a domain at hand are critically important.

Closed-Nature of Deep Blue Work

One criticism of Deep Blue has been that the specific techniques and knowledge used have not been widely reported, making it difficult to adequately assess what is really being demonstrated during these matches. In fact, to many, Deep Blue simply seems synonomous with "the brute-force approach". Much of our knowledge comes from early work on the predecessor, Deep Thought (HACN90) (ACH90).

Furthermore, the number of published games of Deep Blue is relatively very small. All such examples of under-reporting seem to be motivated at least in part by the desire to keep a competitive edge over Kasparov. It would be scientifically far more pleasing for the Deep Blue team, for example, to have the match-ready version of Deep Blue generate hundreds or even thousands of games — perhaps both from self-play and against their chess consultant Grand Master Joel Benjamin — for Kasparov to have in preparation before rematches. Similarly, Deep Blue could prepare (with suitable machine learning) from the huge historic database of Kasparov's games. Presumably, Kasparov's evaluation function and search techniques will not change as rapidly as Deep Blue's might.

Perhaps if Deep Blue wins an entire match against Kasparov, it might become more acceptable to allow such preparations, leading to more significant evaluation of in what ways the human mind may still be superior in the domain of chess. Thus, we argue that once (or if) a computer beats the world champion in a match, perhaps the real science will begin in earnest, although presumably with significantly less public interest and drama.

Summary of Chess-Playing Technologies

In light of the diminishing returns of brute-force search in chess noted above, we have considerable doubts on whether the brute-force style epitomized by Deep Blue will actually suffice to prevail against Kasparov. In this section we highlight a variety of other AI techniques that offer great promise in the longer run.

Search

Perhaps due to the success of Deep Blue, many seem to believe or to assume that parallel algorithms are the prime future technology for computer chess. However, in fact, within the last decade there have been a wide variety of advances in research on game-tree search.

It is important to keep in mind that, due to the design requirements that the Deep Blue hardware places on its search algorithms, it is not clear which of these techniques could be easily incorporated into the Deep Blue approach.

Some elegant advances, though relatively modest performance-wise, have arisen from recent work by Platt et al. Their MTD(f) procedure (PSPdB95) (PSPdB96a) (Pla) essentially reformulates the complex best-first method of SSS* (Sto79) into an elegant combination of zero-window alpha-beta searches and extensive use of transposition (hash) tables. This work has led to modest yet surprising speed improvements for chess (15%) as well as significant clarity about alpha-beta and SSS* variants in general.

Their work on Enhanced Transposition Cutoffs (ETC) (PSPdB96b) takes advantage of the fact that game "trees" are actually graphs. For example, they explore heuristics to maximize reuse of transposition table results, such as checking moves with transposition table entries first under certain conditions, since those might lead to immediate cutoffs. They report that these heuristics lead to searches with 25% fewer nodes.

Less clear at this point, but potentially much more powerful, are methods for selective tree growth. Recent results (BM96) on approximate probabilistic variants of Berliner's well-known B* algorithm, for example, appear promising. B* involves search using intervals to represent position values, instead of points. Search continues until enough information is gathered and propagated up the tree such that the low estimate on the value of one move becomes higher than the high estimate of all alternatives.

Since position evaluators generally only provide "realistic" point estimates, B* uses *null moves* (GC90) (and other heuristics) to generate optimistic and pessimistic values. To avoid the complexity of managing full probabilistic distributions across these intervals, they explored the use of probability distributions that decay linearly from the realistic value to the optimistic value. To handle instability of position evaluations, they use alpha-beta to conduct shallow probe searches (e.g. full 3-ply, plus tactical extensions).

Recently "rational search" methods have been proposed for selective expansion based on Bayesian methods. This approach has been championed by Eric Baum (Bau92) (Bau93) (BS96). These methods attempt to identify at each iteration some small set of nodes whose expansion would be most informative, based on detailed probabilistic distributions of the evaluations of the current leaf nodes. Though general and theoretically sound, use of these techniques is incompatible with alpha-beta cutoffs. This approach has not yet been demonstrated to be competitive with alpha-beta variants in the domain of chess, but certainly some reasonable approximations may prove useful in the future.

Korf and Chickering has proposed an alternative selective search method called "best-first minimax search" (KC94). Essentially, it always expands the leaf of the current principal variation — i.e. the node which determines the minmax value of the root. They argue that it will not generally suffer from excessive exploration of a single path because of the well-known oscillation of values that occurs during minimax search.

Best-first minimax search is relatively simple yet beats fixed-depth alpha-beta search in the game of Othello for comparable-sized trees, for medium depth alpha-beta trees. However, deep fixed-depth alpha-beta search beats it for comparable-sized trees. This suggests that its best use might be as a relatively shallow probe search for method such as B*, although Korf and Chickering did not discuss nor explore that option.

Another promising area is that of generalizing the notion of a transposition table so that each entry contains an equivalence class of positions instead of just one. Ginsberg's work on partition search (Gin96) has developed this idea into the basis of a world-class bridge program. Although he has not developed representations for chess, he notes that this might provide a more formal approach to the specialized though intriguing "method of analogies" (AVAD75).

Knowledge

Knowledge-engineering has long played a role in competitive computer chess, particularly in the areas of endgame table precompilation and manually crafted opening books. Considerable effort typically goes into selecting features and weights for the evaluation function. The requirement of such initial overhead to develop a competitive chess program has perhaps gone a long way towards discouraging most AI researchers from even attempting, and towards computer chess's fall from grace. With the advent of mature search and machine learning techniques, it seems likely that this will change in the near future — particularly if the Deep Blue approach falls short.

Learning

Recent advances in machine learning offer perhaps the best hope for significant near-term advances in computer chess. Recent surveys such as (Fur96) and Jay Scott's active web site at (Sco97) are good summaries of a variety of approaches that have been proposed to date. Unfortunately, most of these approaches have not been within the context of tuning a competitive-level chess program, making their immediate impact difficult to assess.

For example, there has been a fair amount of work on learning chess evaluation functions, with the NeuroChess work (Thr95) being perhaps one of the most interesting.

Furthermore, there has been some work in learning search control heuristics for game-playing, although the more successful ones seem particularly well-suited for games with simplier features, such as Othello (MM94).

It seems fair to say that machine learning to date has not had nearly the impact on computer chess that it could. It seems logical to expect that this situation will change as the impact of diminishing returns in chess search is better understood — particularly when research chess programs mature to the level that such walls are routinely hit.

Summary of Our Chester Program

In this section we briefly summarize our own research on computer chess. Our search engine uses a variant of B* to conduct high-level selective search, MTD(f) to conduct the relatively shallow low-level probes, and extremely fast move generation and transposition table routines, so as to reach a competitive level required for meaningful evaluation. We call this program Chester (Chess Terminator), which reflects our serious but perhaps naive ambitions. The main novelty of our approach is our means of attempting to learn good interval-valued evaluation functions, suitable for B*, as opposed to relying on heuristics such as null-moves to generate the bounds.

The emphasis of our research is on using chess as a domain to help test the generality of our machine learning techniques being developed for real world tasks such as monitoring the NASA Space Shuttle. (DeC97a) In reality, though, the causality of this line of research is not straightforward, since our ideas for learning functions suitable for monitoring tasks were in fact inspired by ideas from B* search.

We believe and hope that such tight synergy between chess and "real world" domains might make serious research on chess more easily justifiable and even encouraged. In the spirit of this workshop, we wish to stress that one of the significances of the GK vs DB matches is the simple fact that there is renewed interested in computer chess and thus it could become a useful forum through which to educate the public about what AI is about as a whole. This gives added urgency to develop more learning-intensive alternatives to the Deep Blue approach, so that we are better able compare and contrast over spectra of approaches, rather than just the human against machine theme per se.

Detailed discussion of our techniques are beyond the scope of this paper. However, we wish to outline our basic ideas here. In our monitoring work, we have developed an approach called Envelope Learning and Monitoring using Error Relaxation (ELMER), which essentially learns high and low limit functions for each sensor, based on historic spacecraft data (DeC96).

Our techniques involve regression using asymmetric cost functions (DeC97b), similarly to the 0-1 loss functions common in classification tasks. However, the key difference is that our cost function is parametric and we search over those parameters and select the settings which give the best fit on cross-validation data.

In this way, we are able to learn envelopes which do not necessarily suffer from classic problems such as variance underestimation (BQ97). In fact, our approach is specifically designed to address cases which are not handled well by the typical approach of learning confidence intervals by first estimating means and then estimating input-conditional variances (e.g. (NW95), (Bis95)). Thus, we can offer better assurance that the bound estimates are not too tight, as required both for B* position evaluations and for spacecraft sensor limits for automated monitoring with few false alarms.

A simple example which illustrates the usefulness of our approach is the task of learning high and low bounds on the complexity of example run times of an algorithm such as quicksort, given as input only the size N of each data set that was sorted. Consider that we wish to use linear regression (for speed and for understandable results) and use the following reasonable set of features for all such tasks: $lgN, N, NlgN, N^2, N^3, 2^N$. With the high and low bounds of quicksort actually being ($O(N^2)$ and O(N lg N) respectively, it turns out to be difficult to capture these bounds using standard mean plus/minus standard deviation approaches. Essentially, the problem is that there are critical missing inputs (namely, the "sortedness" of each data set to be sorted).

We believe that such an approach is likely to prove useful for learning high and low bounds for chess position evaluation functions suitable for B* style selective search. However, we stress that we have not yet had time to try to develop convincing results to support this conjecture. Furthermore, we suspect that good context-sensitive evaluators are likely to require good feature construction techniques, such as the greedy introduction of products that we are currently experimenting with in our spacecraft domains (SSM92).

References

T.S. Anantharaman, M.S. Campbell, and F. Hsu. Singular extensions: Adding selectivity to brute-force searching. *Artificial Intelligence*, 43(1), 1990.

G. Adelson-Velskiy, V. Arlazarov, and M. Donskoy. Some methods of controlling the tree search in chess programs. *Artificial Intelligence*, 6, 1975.

Eric B. Baum. On optimal game tree propagation for imperfect players. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 507–517, 1992.

E. B. Baum. How a bayesian approachs a game like chess. In *Games: Planning and Learning, AAAI Fall Symposium*, 1993.

Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

Hans J. Berliner and Chris McConnell. B* probability based search. *Artificial Intelligence*, 86(1), 1996. (See http:// www.cs.cmu.edu/ afs/cs.cmu.edu/ user/ccm/www/ home.html#B*).

Christopher M. Bishop and Cazhaow S. Qazaz. Regression with input-dependent noise: A bayesian treatment. *Advances in Neural Information Processing Systems 9*, 1997.

E.B. Baum and W.D. Smith. A bayesian approach to game playing. (See http:// www.neci.nj.nec.com/ homepages/ eric/ bayes1.ps), 1996.

Dennis DeCoste. Learning and monitoring with families of high/low envelope functions: Maximizing predictive precision while minimizing false alarms. Technical Report D-13418, Jet Propulsion Laboratory / California Institute of Technology, March 1996.

Dennis DeCoste. Automated learning and monitoring of limit functions. In *Proceedings of the Fourth International Symposium on Artificial Intelligence, Robotics, and Automation for Space*, Japan, July 1997.

Dennis DeCoste. Bounds estimation via regression with asymmetric cost functions. Japan, August 1997. To be presented at IJCAI-97 poster session.

Mikhail V. Donskoy and Jonathan Schaeffer. Perspectives on falling from grace. In *Computers, Chess, and Cognition*. Springer-Verlag, 1990.

Johannes Furnkranz. Machine learning in computer chess: The next generation. *International Computer Chess Association Journal*, 19(3), 1996. (See ftp:// ftp.ai.univie.ac.at/ papers/ oefai-tr-96-11.ps.Z).

G. Goetsch and M. S. Campbell. Experiments with the null-move heuristic. In *Computers, Chess, and Cognition*. Springer-Verlag, 1990.

Matthew L. Ginsberg. Partition search. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 228–233, 1996.

F. Hsu, T.S. Anantharaman, M.S. Campbell, and A. Nowatzyk. Deep thought. In *Computers, Chess, and Cognition*. Springer-Verlag, 1990.

Andreas Junghanns, Jonathan Schaeffer, Mark Brockington, Yngvi Bjornsson, and Tony Marsland. Diminishing returns for additional search in chess. In *Advances in Computer Chess VIII*. (See http:// www.cs.ualberta.ca/ ~ jonathan/Papers/ dim.ps.Z).

Garry Kasparov. The day that i sensed a new kind of intelligence. In *TIME Magazine*. March 25, 1996.

Richard E. Korf and David Maxwell Chickering. Best-first minimax search: Othello results. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1365–1370, 1994.

David E. Moriarty and Risto Miikkulainen. Evolving neural networks to focus minimax search. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1371–1377, 1994.

David A. Nix and Andreas S. Weigend. Learning local error bars for nonlinear regression. *Advances in Neural Information Processing Systems 7*, 1995.

Barney Pell. A strategic metagame player for general chesslike games. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1378–1385, 1994.

Aske Platt. Mtd(f), a new chess algorithm. WWW page: http:// theory.lcs.mit.edu/ ~ plaat/ mtdf.html.

Aske Plaat, Jonathan Schaeffer, Wim Pijls, and Arie de Bruin. Best-first fixed-depth game-tree search in practice. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 273–281, 1995. (See http:// www.cs.ualberta.ca/ ~ aske/ bfmms.ps).

Aske Plaat, Jonathan Schaeffer, Wim Pijls, and Arie de Bruin. Best-first fixed-depth minimax algorithms. *Artificial Intelligence*, 1996. (See http:// www.cs.ualberta.ca/ ~ aske/ aij-final.ps.gz).

Aske Plaat, Jonathan Schaeffer, Wim Pijls, and Arie de Bruin. Exploiting graph properties of game trees. In *Proceedings of the Fifthteen National Conference on Artificial Intelligence*, pages 234–239, 1996. (See http:// www.cs.ualberta.ca/ ~ aske/ AAAI96-final.ps.gz).

Bart Selman, Rodney Brooks, Thomas Dean, Eric Horvitz, Tom Mitchell, and Nils Nilsson. Challenge problems for artificial intelligence. In *Proceedings of the Fifthteen National Conference on Artificial Intelligence*, 1996.

Roger Schank. Where's the AI? In *AI magazine*. Winter 1991.

Jay Scott. Machine learning in games. WWW page: http:// forum.swarthmore.edu/ ~ jay/ learn-game/ index.html, 1997.

Terence D. Sanger, Richard S. Sutton, and Christopher J. Matheus. Iterative construction of sparse polynomial approximations. *Advances in Neural Information Processing Systems 4*, 1992.

G.C. Stockman. A minimax algorithm better than alphabeta? *Artificial Intelligence*, 12(2), 1979.

Sebastian Thrun. Learning to play the game of chess. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 1069–1076. The MIT Press, 1995.

Andreas S. Weigend and Neil A. Gershenfeld, editors. *Time Series Prediction: Forcasting the Future and Understanding the Past*. Addison-Wesley, 1994.