Computers and Chess

Early Artificial Intelligence (AI) researchers were interested in Chess - it required calculation but it also had an element of creativity and intuition. Programs were soon developed using ideas introduced by Shannon and many others to more efficiently prune the move trees. As computing power increased, these programs became capable of beating the best human players. Deep Blue (using specialised hardware) played the world champion Kasparov in 1996, winning a game but losing the match. In 1997 it won a rematch, though this may have had more to do with Kasparov's approach than an improvement by Deep Blue. Even without specialised hardware programs like Fritz running on standard PCs can complete for first place in national championships (Holland 2000, for example). So chess is an AI success story - one of the few early dreams which have come true.

In a way, however, chess has been an AI disappointment. The above programs tend to have a fairly simple static analysis routine. They gain their power by number-crunching through as many positions as they can. They don't "plan", or "learn" in the way that the AI pioneers had expected to be necessary, and their development hasn't led to ideas that have been of wider use.

But there is another parallel strand of chess program development. Botvinnik (ex-World Champion and an Electrical Engineer) amongst others devoted time trying to give computers an "understanding" (in the human sense) of chess positions. This approach has fallen into neglect - it hasn't produced powerful chess programs - but now with more powerful computers and programming techniques it might be time for a revival. Benefits include

- **Improved playing ability** - Though using a pure "understanding" approach may not lead to strong chess performance, a hybrid system might work well. As multi-processor facilities increase, more CPU power will become available. Using all this power for number-crunching may lead to diminishing returns. The influence of the "understanding" module may need to vary according to the type of position. Sometimes it might help sort the candidate moves for the number-crunching, sometimes it may determine to move to play by itself. One model might be the way that in some tournaments players are allowed to use computers to augment their play.

  Also chess isn't just a matter of playing the "best" move. If the program is losing, or if program notices the opponent being cautious, perhaps it should consider complicating the position or trying to set traps. The ability to define moves as traps and assess their likelihood of success would be useful.

- **Annotating** - Many programs offer an option to annotate a game, but most just evaluate the move played and show the best alternative. Published annotations do much more, pointing out what moves are "forced", when the final, critical error was made, and pointing out when an obvious move isn't good. The ability to define moves as "obvious" would be useful. Also few if any programs are able to explain their behaviour in terms that humans would easily comprehend.

  As an exercise it's useful to try to define in computer program terms the criteria for using "!", "!!", "?", "??", "!?" and "?!" in automated annotations. It's not sufficient to say that "!" should be used for a move that leads to a significantly better position than any alternative move would. After all, if someone takes your Queen and you recapture, the recapture is likely to be a significantly better move than the alternatives, but it hardly deserves a "!". Maybe a "!" move is one that in the short term doesn't produce an improved position (indeed, if it's a sacrifice it produces a worse one) but in the long term does improve the position.

When the value of a move changes suddenly as the depth of analysis increases, I think one could say that the move is in some sense "interesting" - oversights, traps and sacrifices fall into this category.

- **Training** - Computers will be more useful as training tools if they play more like humans - some programs are much better at passing the "Turing Test" than others (by the way, Turing was a keen though not very good player and is credited with having written the first computer chess program in 1950). Such programs may also be able diagnose human weaknesses and offer appropriate remedial exercises.
- **Psychology insights** - Chess has been used as a tool by psychologists. The number-crunching approach has provided few insights.

Adding some of these facilities to freely available chess programs as modules would be an interesting project.

See also

- chess notes for more detailed suggestions.
- the annotations page for more detailed suggestions about annotating.

References

- "Computers, Chess, and Cognition", Edited by T. Anthony Marsland and Jonathan Schaeffer, Springer-Verlag, ISBN 0-387-97415-6, 1990.
- International Computer Chess Association
- crafty (current work includes improving the chess knowledge contained in the program so that it plays better positional chess and also so that its strategy is goal-oriented rather than random). See also the Crafty HomePage
- GNU chess (Positional learning, book learning, and temporal difference learning are being experimented with but are not in v5.00 and are not guaranteed to be in any future version).
- Further notes on computer chess (by Dr A. N. Walker)
- "Using patterns and plans in chess", Wilkens, David, Artificial Intelligence, vol. 14 (1980), 165-203.
- "Experienced-based creativity", Levinson, Robert, Dept. of Computer and Info. Sciences, Univ. of California Santa Cruz, undated unpublished manuscript, 36 pages.
- "Reasoning with uncertainty in computer chess", Horacek, Helmut, Artificial Intelligence, vol. 43 (1990), 37-56. (V43 is a special issue on Computer Chess).
- "Thought and choice in chess", De Groot, Mouton & Co, 1965.
- "A program for learning to play chess", Pitrat, J., in "Pattern Recognition and Artificial Intelligence", Chen, C.H. (ed), 1976
- "Using Plans in a Chess Playing Program", Pitrat, J., 5th IJCAI, pp 979-982, 1977.
- "Changes in Representation which Preserve Strategies in Games", Banerji, H.B. and Ernst, G.W., in Proc. 2nd IJCAI, pp 651-658.
- "Similarities in Games and Their Use in Strategy Construction", Banerji, H.B., in "Computers and Automata Proceedings of 21st Brooklyn Polytechnic Symposium, pp 337-357", 1971
- "Computer Chess Compendium", David Levy, Batsford, 1998.
- The rec.games.chess.computer newsgroup
- Computer Chess Publications (Paul Verhelst)
- "Computers, Chess and Long Range Planning", Botvinnik, M.M., Longman, 1970. (*he presents an algorithm that 'represents the thought processes of a chess master during the game'*)
- Annotated bibliography of Othello programming

- "A Model of Visual Organisation for the Game of Go", Zobrist, A.L., Proc SJCC 69, pp 103-112, 1969.
- Dragoon Chess
- Searching for Solutions in Games and Artificial Intelligence by L. Victor Allis.
- "Advanced in Computer Chess", Beale (issues 1-6 are in Cambridge's UL at 414:01.b.1-6 on SF6).

Updated: October 2002

---

tpl@eng.cam.ac.uk