Tournament Competition Fuels Computer Chess

Chess programs that win do not try to think like a grand master; in fact, they know relatively little about chess at all

The din in the packed arena had hardly subsided to a mild roar, when suddenly the fans exploded once again. "What a move!" screamed one nearby spectator. "How did he ever find a way through that defense?" wondered another, overcome with admiration.

With only a touch of exaggeration, this scene applies as well to any recent computer chess tournament as it does to last month's NBA basketball play-offs. In short, because of the head-to-head competition, computer chess is distinctly different from other fields of research. Moreover, under the pressure of tournament competition, the desire to win has become so strong that it is fixing the direction of research. In this way, computer chess, which has been described as the *Drosophila* (fruit fly) of artificial intelligence, has drifted away from what some see as its most important role—a vehicle for aiding scientists in discovering how people think and make decisions and, eventually, how to simulate these processes in "intelligent" machines.

Not everyone, of course, sees the effects of tournament competition as all bad. Some observers regard as a misconception the idea that intelligent computers must become that way by simulating humans. In this view, computer chess tournaments have helped immensely by ramming home the message that the most successful chess programs are those that take advantage of what today's computers do best—calculate or manipulate data very rapidly.

Another beneficial effect of computer chess tournaments is the motivation they have given programmers to improve their strategies over the years. In this age of Golden Fleece awards, federal agencies can be excused for not supporting such apparently frivolous ventures as computer chess. But, notes Hans Berliner of Carnegie-Mellon University, computer chess programs are more competent in what they do than any other part of artificial intelligence because there has been a continuous, although largely unfunded, effort over the last decade.

Chess, in common with such serious enterprises as modeling international conflicts or making decisions about the economy, brings the participants face to face with a situation of the form "If I do this, he'll counter with that, and I'll come back with this, and he'll answer me with that, and so forth . . ." in which the object is to make the best possible decision. On the face of it, the best move will be made by the person who can examine the largest number of such sequences and can follow them the farthest. In accord with this supposition, there is a mystique about chess that the best chess players are geniuses with astronomical IQ's and prodigious memories who can therefore investigate numerous sequences of moves far into the future. If this is what it takes to play good chess, then a computer should make an ideal chess player.

One problem with investigating every possible move, first pointed out by Claude Shannon 30 years ago in his discussion of possible approaches to computer chess, is that there are on the average about 35 moves a chess player must choose between. The opponent can, again on the average, make 35 responses to each of these 35 moves, and so forth. In this way, a "game tree" is quickly built up with 35^N branches, where N is the "depth" of the search—that is, the number of moves ahead of the present one. In a tournament, humans or computers have only about 3 minutes (another average) for each move. To evaluate the worth of each of these potential moves is clearly impossible for even the fastest computer when N increases beyond about 6. Since an average game between chess masters lasts about 84 moves (42 for each side), computers really cannot see very far ahead in a game.

An obvious way to "prune" the branches of the game tree and reduce the amount of searching is to somehow select only the most promising moves for further evaluation. The most successful early computer chess programs, notably that of Richard Greenblatt of the Massachusetts Institute of Technology, which appeared in 1966 (Elo rating* of 1400), adopted this approach. Greenbatt's program selected promising moves by means of a plausible move generator, which is a section of the program containing information on chess and instructions for using this information to generate moves. Greenblatt says that his program could play a creditable game of chess for long stretches of time but would then follow with a very poor move. Often, such blunders, which are the occasion for choruses of boos and catcalls from the vocal audiences at computer tournaments, are serious enough to lose the game to a competent human opponent.

Knowledgeable chess programmers realized all along, in part through the work of Alan Newell and Herbert Simon of Carnegie-Mellon in the 1950's and the studies of Adriaan de Groot in the Netherlands and Soviet researchers even earlier, that chess masters actually do not function in the manner prescribed by the chess mystique. In fact, chess masters investigate rather few sequences of moves but probe these very carefully. Moreover, the selection of promising lines of play seems to be largely an intuitive act based on the recognition of the overall patterns of chess positions that have been encountered in previous matches or studied in chess manuals. In intelligence and ability to memorize large quantities of detailed information, chess masters are not distinguishable from comparably educated nonchess players.

Perhaps it was the realization that hu-

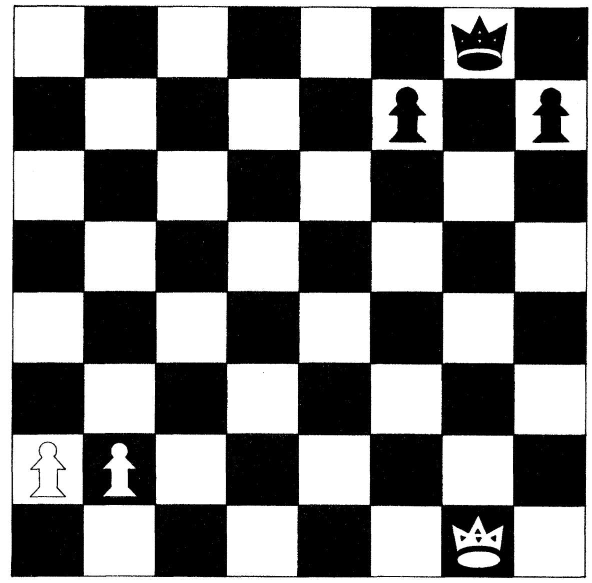
*Chess players, and now computer chess programs, by competing in tournaments regularly, gain a rating. A refinement of this rating system, developed by Arpad Elo of Marquette University, quantitatively measures the probability of one player defeating another. The United States Chess Federation classes (and their Elo numbers) are: E (less than 1200), D (1200 to 1399), C (1400 to 1599), B (1600 to 1799), A (1800 to 1999), expert (2000 to 2199), master (2200 to 2399), and senior master (above 2400). The Fédération Internationale des Echecs (FIDE) has additional titles: international master (2300 to 2499) and international grand master (2500 and above). The better player will win 75 percent of the time, when the Elo numbers of two competitors differ by 200 points. The median of American tournament players is said to be about 1400.

man chess players do not rely solely on mechanical computation or memorization skills that led to linking computer chess with artificial intelligence. To develop a chess program that could compete with the masters, it would be necessary to duplicate in the computer the processes that went on in the masters' heads. In any case, artificial intelligencers did not grasp how difficult a task this would be, and, in a now celebrated 1968 wager, two of them (Donald Michie of the University of Edinburgh and John McCarthy of Stanford University) each bet Scottish international master David Levy £250 that a computer could defeat him in chess within 10 years. Three years later, two others had joined in the bet against Levy (Seymour Papert of MIT—£250, and Ed Kozdrowicki of the University of California at Davis—$650). Levy collected most of his money last August, when the bet expired and no computer had gotten more than 1.5 points in a 5-point match with him.

[Levy's wager and the excitement it caused have begun a trend. There is now a proliferating number of bets and prizes for programs that can defeat chess masters. A Dutch software company (Volmac) recently offered, for example, the sizable sum of $50,000 to anyone who can write a computer program that could beat former world champion and international grand master Max Euwe of the Netherlands by 1 January 1984.]

In the meantime, the first North American Computer Chess Championship, organized by Monroe Newborn (now at McGill University) was held in 1970 during the New York City meeting of the Association for Computing Machinery (ACM). Tournaments have been held every year since. And, under the auspices of the International Federation for Information Processing, world championships were held in 1974 (Stockholm) and 1977 (Toronto). Although it took a couple of years, these tournaments seem to have been instrumental in setting computer chess on its present course away from the artificial intelligence ideal of simulating a human chess player.

The winner of the first North American tournament was a program called Chess 3.0 written by David Slate, Larry Atkin, and Keith Gorlen of Northwestern University, which went undefeated in the tournament. Although Chess 3.0 was a successful example of Greenblatt's approach to computer chess (it won again in 1971 and 1972), in 1973 Slate and Atkin extensively revised their program and rechristened it Chess 4. Although it lost to a Soviet program called Kaissa in the 1974 World Championship, Chess 4.6 (an improved version of Chess 4) wrested the world crown back in 1977 and is the reigning computer chess king until the upcoming world tournament next year in Melbourne.



One limitation of full-width search methods in computer chess is called the horizon effect. A human easily sees that white, although it has one less pawn, can win the game by advancing its leftmost pawn across the board, whereupon it is promoted, by the rules of chess, to a queen. The computer, if it does not search enough moves into the future, will not see this outcome and will be excessively concerned about black's apparent material advantage. [Drawing by Eleanor Warner]

What was shocking, in its way, was that the Chess 4 series of programs reverted to the original conception of how a computer should play chess. The program did not use chess knowledge to choose plausible moves. Its strength was its ability to rapidly search *every* possible move to a certain depth in the game tree, so-called full-width or exhaustive searching, as opposed to the narrow-width or selective searching of the Greenblatt program. The only computer chess programs that have beaten Chess 4.6 or its successor, Chess 4.7, in the last 2 years (Duchess by Tom Truscott, Eric Jensen, and Bruce Wright of Duke University, and Belle by Ken Thompson and Joe Condon of Bell Laboratories) are of the same type and are sometimes spoken of as clones of the Northwestern program because they are so similar. Their similarities extend to their histories; researchers at both Duke and Bell Laboratories experimented with selective search programs before following the Northwestern lead and moving to a full-width search.

These three programs are, as judged by their Elo rating, highly successful. Taken at face value, the Northwestern rating (2040), the Bell Laboratories rating (estimated at 2000), and the Duke rating (1890), suggest that the best computer chess programs are nearing the expert level. A sanguine judgment of the future held by some is that, as computers get faster and faster, the capability of these types of chess programs will gradually improve to the point they will be able to defeat the best human players, but no one is sticking his neck out by predicting when this will come about.

The method used by these programs, according to Slate, has the rather hair-raising name of progressively deepened, alpha-beta, depth first, minimax search with quiescence searching of unstable positions. Minimax is a strategy for evaluating positions in the game tree traceable back to the 1944 book by John von Neumann and Oskar Morgenstern on game theory. The alpha-beta algorithm is a procedure, originally proposed more than 20 years ago by Newell, Simon, and J. Clifford Shaw of the Rand Corporation, that obviates the need for generating and evaluating every possible position in the game tree, and in this way considerably reduces the computation required. The other features, of more recent vintage, further increase the efficiency of the search process.

Not surprisingly, the success of such full-width search programs depends not only on the efficiency of the program that implements it but also on the computer itself. In essence, a faster machine can search farther in a given time, and a deeper search is likely to result in a better move. (This intuitively obvious conclusion is not always correct. Dana Nau of Duke has recently shown that under certain conditions, deeper searches decrease the probability of making the right move. These conditions are apparently rare in chess, however.) Thus, the tendency has been to use the largest computers available. Barend Swets of the International Institute for Hydraulic and Environmental Engineering in Delft, Netherlands, who participated in the 1977 Toronto tournament, estimated that the value of the computers used by the 16 entrants surpassed $100 million.

Nonetheless, super computers are not

necessarily the wave of the future. At the last North American tournament, held in December in Washington, D.C., Dan and Kathe Spracklen of San Diego finished in a tie for third in a 12-contestant field. The Spracklen's program, Sargon II, ran on a microcomputer, but still managed to incorporate many of the features of programs run on larger machines. The main difficulty with microcomputers, says Spracklen, is not speed, although they are slower, but the lack of a large memory. This deficiency so far has prevented adding features, such as a table of already evaluated positions, that require storing much information. Sargon II will soon be available in the form of a commercial chess-playing computer similar to those now on the market for about $100.

Perhaps an even more dramatic piece of evidence against the need for large computers is the program of Thompson and Condon at Bell Laboratories, which won the Washington tournament. Belle runs on a minicomputer at a cost, Thompson estimates, of $1.50 per hour as compared to typical charges of $1000 or more per hour for large computers. (The Northwestern team has a unique arrangement with the Control Data Corporation, which allows it free use of a Cyber 176 computer that can evaluate about 3600 positions per second during tournaments and exhibitions.)

Thompson's secret is to forego some flexibility for a great deal of speed incorporating some nonprogrammable (hardwired) modules designed specifically to carry out chess calculations. There are modules to generate all legal chess moves, execute the moves, evalute the moves, and store 2000 of the most recently evaluated positions and their scores. By, in effect, building a special-purpose computer, Thompson calculates he can evaluate twice as many positions per second as Northwestern. And with attention to using special high-speed microelectronic circuits, he thinks the speed could be pumped by another factor of 10, which is thought by some to be equivalent to about 200 more Elo points.

What Thompson's success will lead to is unclear. Some observers are not convinced that Belle's win over Chess 4.7 at Washington was representative of the strengths of the two programs. Hans Berliner of Carnegie-Mellon believes, however, that there will a rush to copy Thompson's approach as the improved version starts to accumulate victories. Fred Swartz of the University of Michigan also thinks Belle will be highly successful but reaches a different conclusion. Says Swartz, Thompson will stimulate a return to the selective search method because no one will be able to beat him in a full-width search.

Swartz and his colleagues at Michigan have been supporters of a selective search approach since 1972, so it is not true that the full-width search is the only credible method, despite its great success. As compared to the latter search method, which may evaluate several hundred thousand positions in the 3 minutes allowed per move in a tournament, Swartz's program, Chaos, looks at only about 30,000 positions before making a move. (An even more selective program, Awit, by Tony Marsland of the University of Alberta, seldom investigates more than 500 positions.)

To compensate for the possibility of making a particularly poor move or of overlooking an obviously good one, Swartz must incorporate much more chess knowledge into his program than the full-width searchers do. Thus, the program is more complicated, takes more computer memory, and runs more slowly. Although many fewer positions are evaluated, it takes longer to do the evaluation of each position, and both full-width and selective searches penetrate to about the same depth in the game tree.

In the Washington tournament, Chaos (Elo rating of 1800) almost defeated Thompson's program Belle; it had a winning position. But just as Greenblatt's early program was plagued by sudden lapses of intelligence, so was Chaos. A blunder permitted Belle to win.

To guard against this happening, in the future, Greenblatt is now investigating a hybrid approach. Some years ago, he built a machine that was exceptionally dumb about chess but super fast (100,000 positions per second) to conduct full-width searches of game trees. Greenblatt's idea is to combine this high-speed machine, which he calls Cheops, with a new selective search program still under development at MIT. In this scheme, the basic play selection will be the responsibility of the new program but Cheops will stand guard against blatant sins of commission or omission.

Carnegie-Mellon's Berliner would like to build a chess-playing program with even more chess knowledge because, in his view, there are inherent limitations in an approach that only looks ahead a certain number of moves, makes a crude evaluation of the situation, and has no comprehension of what is happening. Although recognizing that a successful program may lie many years away, Berliner points to the recent work of David Wilkins of Stanford for an example of how such a program might function.

Wilkins has devised a program that plays the middle stages of a chess game; it cannot handle opening sequences of moves nor the final moves (end game) when few pieces remain on the board. It reverses the usual computer chess strategy of using tree searching to select the best move. Wilkins' program, Paradise, uses chess knowledge incorporated into the program to devise a plan of action. Tree searching is limited to verifying that the move suggested by the plan is in fact the best move, and typically only about 40 positions are examined in this process. Formation of the plan is by way of pattern recognition. A set of 200 production rules (portions of the program) look for patterns on the chessboard. When the pattern sought by a particular rule is found, another part of the program called a knowledge source is activated. Each knowledge source provides the information necessary for the computer to understand and reason about a particular chess concept. The program then links those concepts activated by the pattern matching to create a plan of action. Wilkins says that in a test composed of 100 problems from a well-known chess book, his program scored better than 97 percent correct moves.

While Wilkins' approach seems to be leading back to the idea that chess is a useful vehicle for eliciting information about human thinking processes and for simulating these in a computer, observers still do not agree whether or not this is the role that computer chess is best suited to play. It may be that computers are best used in a mode that takes direct advantage of their computational speed and remarkable memory, a view espoused by Alan Biermann of Duke. For example, Thompson has constructed a program that could play a well-known end game (queen and king versus rook and king). The best the side with the rook can do is stall for a draw (after 50 moves without a capture, a draw is called). In a demonstration against chess masters, including Berliner and Walter Browne, the top American chess player, Thompson's program was never drawn when it had the queen and never beaten when it had the rook—not that is, until Browne had a week to study the situation and discovered an aspect of end-game strategy that was not appreciated by humans until the computer demonstrated it to them. The machine, of course, knew nothing about the strategy; it simply had stored, in a giant table, the precalculated outcomes of all possible moves and it played accordingly.

—Arthur L. Robinson