Computer chess

From Wikipedia, the free encyclopedia


1990s Pressure-sensory Chess Computer with LCD screen

The idea of creating a chess-playing machine dates back to the eighteenth century. Around 1769, the chess playing automaton called The Turk became famous before being exposed as a hoax. Serious trials based in automatons such as El Ajedrecista were too complex and limited to be useful.

The field of mechanical chess research languished until the advent of the digital computer in the 1950s. Since then, chess enthusiasts and computer engineers have built, with increasing degrees of seriousness and success, chess-playing machines and computer programs.

Chess-playing computers are available for negligible cost, and there are many programs (such as free software, like GNU Chess, Amy and Crafty) that play a game that, with the aid of virtually any modern personal computer can defeat most master players under tournament conditions, while top commercial programs like Shredder or Fritz have surpassed even world champion caliber players at blitz and short time controls.

Contents


Chessmaster 10th edition running on Windows XP


3-D board, Fritz 8

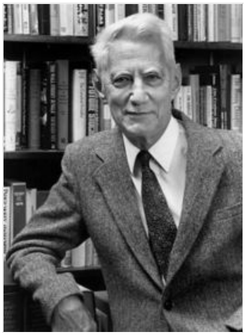
Motivation

The prime motivations for computerized chess playing have been solo entertainment (allowing players to practice and to amuse themselves when no human opponents are available), as aids to chess analysis, for computer chess competitions, and as research to provide insights into human cognition. For the first two purposes computer chess has been a phenomenal success — going from the earliest real attempts to programs which challenge the best human players took less than fifty years. We can say that chess play is not an intractable problem to modern computing.

However, to the surprise and disappointment of many, chess has taught us little about building machines that offer human-like intelligence, or indeed do anything except play excellent chess. For this reason, computer chess, (as with other games, like Scrabble) is no longer of great academic interest to researchers in artificial intelligence, and has largely been replaced by more intuitive games such as Go as a testing paradigm. Chess-playing programs essentially explore huge numbers of potential future moves by both players and apply a relatively simple evaluation function to the positions that result, whereas computer Go challenges programmers to consider conceptual approaches to play.

Brute-force methods are useless for most other problems artificial intelligence researchers have tackled, and are very different from how human chess players select their moves. In some strategy games, computers easily win every game, while in others they are regularly beaten even by amateurs. In chess, the combined skills of knowledgeable humans and computer chess engines can produce a result stronger than either alone.

Brute force versus selective search


Claude Shannon

The first paper on the subject by Claude Shannon, published in 1950 before anyone had programmed a computer to play chess, successfully predicted the two main possible search strategies which would be used, which he labeled 'Type A' and 'Type B' (Shannon 1950).

Type A programs would use a "brute force search" approach, examining every possible position for a fixed number of moves using the minimax algorithm. Shannon believed this would be impractical for two reasons.

First, with approximately thirty moves possible in a typical real-life position, he expected that searching the approximately 30^6 (over 700,000,000) positions involved in looking three moves ahead for both side (six plies) would take about sixteen minutes, even in the "very optimistic" case that the program evaluated a million positions every second. (It took about forty years to achieve this speed.)

Second, it ignored the problem of quiescence, trying to only evaluate a position that is at the end of an exchange of pieces or other important sequence of moves ('lines'). He expected that adapting type A to cope with this would greatly increase the number of positions needing to be looked at and slow the program down still further.

Instead of wasting processing power examining bad or trivial moves, Shannon suggested that type B programs would use a "strategic AI (Artificial Intelligence)" approach to solve this problem by only looking at a few good moves for each position. This would enable them to look further ahead ('deeper') at the most significant lines in a reasonable time.

Adriaan de Groot interviewed a number of chess players of varying strengths, and concluded that both masters and beginners look at around forty to fifty positions before deciding which move to play. What makes the former much better players is that they use pattern recognition skills built from experience. This enables them to examine some lines in much greater depth than others by simply not considering moves they can assume to be poor.

More evidence for this being the case is the way that good human players find it much easier to recall positions from genuine chess games, breaking them down into a small number of recognizable sub-positions, than completely random arrangements of the same pieces. In contrast, poor players have the same level of recall for both.

The problem with type B is that it relies on the program being able to decide which moves are good enough to be worthy of consideration ('plausible') in any given position and this proved to be a much harder problem to solve than speeding up type A searches with superior hardware.

One milestone was the abandonment of type B in 1973 by the team from Northwestern University responsible for the Chess series of programs, who had won the first three ACM Computer Chess Championships (1970-72). The resulting program, Chess 4.0, won that year's championship and its successors went on to come second in both the 1974 ACM Championship and that year's inaugural World Computer Chess Championship, before winning the ACM Championship again in 1975, 1976 and 1977.

One reason they gave for the switch was that they found it less stressful during competition, because it was difficult to anticipate which moves their type B programs would play, and why. They also reported that type A was much easier to debug in the four months they had available and turned out to be just as fast: in the time it used to take to decide which moves were worthy of being searched, it was possible just to search all of them.

Chess 4.0 type programs won out for the simple reason that their programs simply played better chess. Such programs did not try to mimic human thought processes, but relied on full width alpha-beta/negascout searches. Most such programs (including all modern programs today) also included a fairly limited *selective* part of the search based around quiescence searches, and usually extensions and pruning (particularly null move pruning from the 1990s onwards) which were triggered based on certain conditions in and attempt to weed out or reduce obviously bad moves (history moves) or to investigate interesting nodes (e.g. check extensions, passed pawns on seventh rank, etc). Extension and pruning triggers have to be used very carefully however. Over extend and the program wastes too much time looking at uninteresting positions. If too much is pruned, there is a risk cutting out interesting modes. Chess programs differ in terms of how and what types of pruning and extension rules are included as well as in the evaluation function. Some programs are believed to be over selective than others (for example Deep Blue was known to be less selective than most commercial programs because they could afford to do more complete full width searches), but all have a base full width search as a foundation and all have some selective components (Q-search, pruning/extensions).

Though such additions meant that the program did not truly examine every node within its search depth (so it would not be truly brute force in that sense), the rare mistakes due to these selective search was found to be worth the extra time it saved. so it could search deeper. In that way Chess programs can get the best of both worlds.

Furthermore, technological advances by orders of magnitude in processing power have made the brute force approach far more incisive than was the case in the early years. The result is that a very solid, tactical AI player aided by some limited positional knowledge built in by the evaluation function and pruning/extension rules began to match the best players in the world. It turned out to produce excellent results, at least in the field of chess, to let computers do what they do best (calculate) rather than coax them into imitating human thought processes and knowledge. In 1997 Deep Blue, defeated World Champion Garry Kasparov, marking the first time a computer has defeated a reigning world chess champion in standard time control.

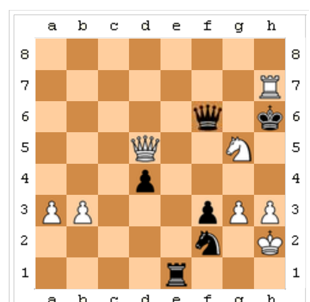
Computers versus humans

However for a long time in the 1970s and 1980s it remained an open question whether any Chess program would ever be able to defeat the expertise of top humans. In 1968, International Master David Levy made a famous bet that no chess computer would be able to beat him within ten years. He won his bet in 1978 by beating Chess 4.7 (the strongest computer at the time), but acknowledged then that it would not be long before he would be surpassed. In 1989, Levy was crushed by the computer Deep Thought in an exhibition match.

Kasparov vs. Deep Blue

Deep Thought, however, was still considerably below World Championship Level, as the then reigning world champion Garry Kasparov demonstrated in two sterling wins in 1989. It wasn't until a 1996 match with IBM's Deep Blue that Kasparov lost his first game to a computer at tournament time controls in Deep Blue - Kasparov, 1996, Game 1. This game was, in fact, the first time a reigning world champion had lost to a computer using regular time controls. However, Kasparov regrouped to win three and draw two of the remaining five games of the match, for a convincing victory.

In May 1997, an updated version of Deep Blue defeated Kasparov 3½-2½ in a return match. IBM keeps a web site of the event (http://www.chess.ibm.com) . While not an official world championship, the outcome of the match is often taken to mean that the strongest player in the world is a computer. Such a claim is open to strong debate, as a truly fair human-machine match is difficult to arrange. Seen as unfair is that human players must win their title in tournaments which pit them against a diverse set of opponents' styles, while computers are occasionally optimized for the their current opponent. Also computers have access to huge databases for opening and end play, something which their human opponent miss.



The final position of Game One,

Deep Blue vs. Kasparov 1996

IBM retired Deep Blue after the match and it has not played since. However, other "Man vs. Machine" matches continue to be played.

With increasing processing power, Chess programs running on humble workstations began to rival top flight players. In 1998, Rebel 10, defeated Viswanathan Anand who at the time was ranked second in the world, by a score of 5-3. But one should note that out of the four blitz games (five minutes plus five seconds Fischer delay (see time control) for each move), Rebel won 3-1, in the two semi-blitz games (fifteen minutes for each side) Rebel won 1½-½ and in the tournament game (forty moves in two hours, one hour sudden death), it was Anand who won. Noteworthy is that computers play better at faster time controls, but a player's actual strength is only determined at the classical time controls, where Anand proved that humans were still better [1]. In the early 2000s, commercially available programs such as Deep Junior and Fritz were able to draw matches against former world champion Garry Kasparov and former "classical" world champion Vladimir Kramnik.

In October 2002, Vladimir Kramnik and Deep Fritz competed in the eight-game Brains in Bahrain match, which ended in a draw. Kramnik won games 2 and 3 by "conventional" anti-computer tactics—play conservatively for a long-term advantage the computer is not able to see in its game tree search. Fritz, however, won game 5 after a severe blunder by Kramnik. Game 6 was described by the tournament commentators as "spectacular." Kramnik, in a better position in the early middlegame, tried a piece sacrifice to achieve a strong tactical attack, a strategy known to be highly risky against computers who are at their strongest defending against such attacks. True to form, Fritz found a watertight defence and Kramnik's attack petered out leaving him in a bad position. Kramnik resigned the game, believing the position lost. However, post-game human and computer analysis has shown that the Fritz program was unlikely to have been able to force a win and Kramnik effectively sacrificed a drawn position. The final two games were draws. Given the circumstances, most commentators still rate Kramnik the stronger player in the match.

- In January 2003, Garry Kasparov played Deep Junior, another chess computer program, in New York. The match ended 3-3.
- In November 2003, Garry Kasparov played X3D Fritz. The match ended 2-2.

Grandmaster Michael Adams

In 2005, Hydra, a dedicated chess computer with custom hardware and sixty-four processors and also winner of the 14th IPCCC in 2005, crushed seventh-ranked Michael Adams 5½-½ in a recent six-game match (though Adams' preparation was far less thorough than Kramnik's 2002 series). Some commentators [2] believe that Hydra will ultimately prove clearly superior to the very best human players, or if not its direct successor will.

In November-December 2006, Vladimir Kramnik played Deep Fritz. This time the computer won, the match ended 2-4.

Endgame tablebases

Main article: Endgame tablebase

Computers have been used to analyze some chess endgame positions completely. Such endgame databases are generated in advance using a form of retrograde analysis, starting with positions where the final result is known (e.g. where one side has been mated) and seeing which other positions are one move away from them, then which are one move from those etc. Ken Thompson, perhaps better known as the key designer of the UNIX operating system, was a pioneer in this area.

Endgame play had long been one of the great weaknesses of chess programs because of the depth of search needed, with some otherwise master-level programs being unable to win in positions that even intermediate human players would be able to force a win.

The results of the computer analysis sometimes surprised people. In 1977 Thompson's Belle chess machine used the endgame tablebase for a king and rook against king and queen and was able to draw that theoretically lost ending against several masters (see Philidor position#Queen versus rook). This was despite not following the usual strategy to delay defeat of keeping the defending king and rook close together for as long as possible. Asked to explain the reasons behind some of the program's moves, Thompson was unable to do so beyond saying the program's database simply evaluated its moves as best.

Most grandmasters declined to play against the computer in the queen versus rook endgame, but Walter Browne accepted the challenge. A queen versus rook position was set up in which the queen can win in thirty moves, with perfect play. Browne was allowed 2½ hours to play fifty moves, otherwise a draw would be claimed under the fifty move rule. After forty-five moves, Browne agreed to a draw, being unable to force checkmate or win the rook within the next five moves. In the final position, Browne was still seventeen moves away from checkmate, but not quite that far away from winning the rook. Browne studied the endgame, and played the computer again a week later in a different position in which the queen can win in thirty moves. This time, he captured the rook on the fiftieth move, giving him a winning position (Levy & Newborn 1991:144-48), (Nunn 2002:49).

Other positions, long believed to be won, turned out to take more moves against perfect play to actually win than were allowed by chess's fifty move rule. As a consequence, for some years the official laws of chess were changed to extend the number of moves allowed in these endings. After a while, the law reverted back to fifty moves in all positions — more such positions were discovered, complicating the rule still further, and it made no difference in human play, as they could not play the positions perfectly.

Over the years, other endgame database formats have been released including the Edward Tablebases, the De Koning Endgame Database (released in 2002) and the Nalimov Endgame Tablebases which is the current standard supported by most chess programs such as Shredder or Fritz. All endgames with five or or fewer pieces have been analyzed completely. Of endgames with six men all positions have been analyzed except for positions with five pieces against a lone king [3]. Some seven-piece endgames, have been analyzed by Marc Bourzutschky and Yakov Konoval [4]. In all of these endgame databases it is assumed that castling is no longer possible.

The databases are generated by storing in memory the values of positions which have been encountered so far, and using these results to lop off the ends of the search trees if they arise again. Although the number of possible games after a number of moves rises exponentially with the number of moves, the number of possible positions with a few pieces is exponential only in the number of pieces — and effectively limited however many end game moves are searched. The simple expediency of remembering the value of all previously reached positions means that the limiting factor in solving end games is simply the amount of memory available in the computer. While computer memory sizes are increasing exponentially, there is no reason why end games of increasing complexity should not continue to be solved.

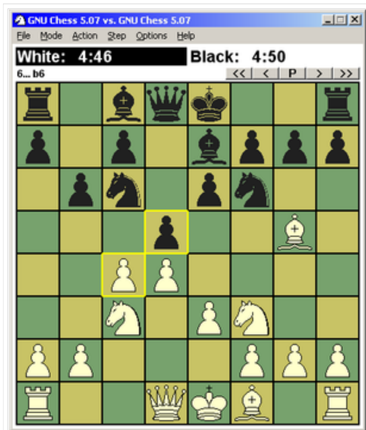
A computer using these databases will, upon reaching a position in them, be able to play perfectly, and immediately determine whether the position is a win, loss or draw, plus the fastest way of getting to that result. Knowledge of whether a position is a win, loss or draw is also helpful in advance since this can help the computer avoid or head towards such positions depending on the situation.

Endgame databases featured prominently in 1999, when Kasparov played an exhibition match on the Internet against the Rest of the World. A seven piece Queen and pawn endgame was reached with the World Team fighting to salvage a draw. Eugene Nalimov helped by generating the six piece ending tablebase where both sides had two Queens which was used heavily to aid analysis by both sides.

Computer chess implementation issues

Developers of chess-playing computer system must decide on a number of fundamental implementation issues. These include

- Board representation (how a single position is represented in data structures).
- Search techniques (how to identify the possible moves and select the most promising ones for further examination),
- Leaf evaluation (how to evaluate the value of a board position, if no further search will be done from that position).



GNU Chess 5.07 on WinBoard 4.2.7

Implementors also need to decide if they will use endgame databases or other optimizations, and often implement common *de facto* chess standards.

Board representations

The data structure used to represent each chess position is key to the performance of the application. Methods range from obvious and easy (an array) to obscure and difficult Bitboard.

Search techniques

Computer chess programs consider chess moves as a game tree. In theory, they examine all moves, then all counter-moves to those moves, then all moves countering them, and so on, where each individual move by one player is called a "ply". This evaluation continues until it reaches a final "leaf" position which is evaluated.

A naïve implementation of this approach, however, would never finish in a practical amount of time, so various methods have been devised to greatly speed the search for good moves.

For more information, see:

- Minimax algorithm
- alpha-beta pruning
- Killer heuristic
- Iterative deepening depth-first search
- Null-move heuristic

Leaf evaluation

For most chess positions, computers cannot look ahead to all final possible positions. Instead, they must look ahead a few ply and then evaluate the final board position. The algorithm that evaluates final board positions is termed the "evaluation function", and these algorithms are often vastly different between different chess programs.

Evaluation functions typically evaluate positions in hundredths of a pawn, and consider material value along with other factors affecting the strength of each side. When counting up the material for each side, typical values for pieces are 1 point for a pawn, 3 points for a knight or bishop, 5 points for a rook, and 9 points for a queen. (See chess piece point value.) By convention, a positive evaluation favors White, and a negative evaluation favors Black.

The king is sometimes given an arbitrary high value such as 200 points (Shannon's paper) or 1,000,000,000 points (1961 USSR program) to ensure that a checkmate outweighs all other factors (Levy & Newborn 1991:45). Evaluation functions take many factors into account, such as pawn structure, the fact that a pair of bishops are usually worth more, centralized pieces are worth more, and so on. The protection of kings is usually considered, as well as the phase of the game (opening, middle or endgame).

See Claude Elwood Shannon for a description of his early paper about a chess-playing program.

Using endgame databases

Some computer chess operators have pointed out that endgame tablebases have the potential to weaken performance strength in chess computers if incorrectly used. Because some positions are analyzed as forced wins for one side, the program will avoid the losing side of positions at all costs. However, many endgames are forced wins only with flawless play, where an even slight error would produce a different result. Consequently, most modern engines will play many endgames well enough on their own. A symptom of this problem is that computers may resign too early because they see that they are being forced into a position that is theoretically dead lost (although they may be thirty or more moves away from the end of the game, and most human opponents would find it hard to win in that time). This observation is only relevant when a computer program is in a situation where it has a choice between two losing moves, one of which is actually more difficult for the opponent, but leads to a tablebase position with a known value, and is hence of very minor importance.

The Nalimov tablebases do not consider the fifty move rule, under which a game where fifty moves pass without a capture or pawn move can be claimed to be a draw by either player. This results in the tablebase returning results such as "Forced mate in sixty-six moves" in some positions which would actually be drawn because of the fifty move rule. However, a correctly programmed engine does know about the fifty move rule, and in any case if using an endgame tablebase will choose the move that leads to the quickest win (even if it would fall foul of the fifty move rule with perfect play). If playing an opponent not using a tablebase, such a choice will give good chances of winning within fifty moves.

One reason for this is that if the rules of chess were to be changed once more, giving more time to win such positions, it will not be necessary to regenerate all the tablebases. It is also very easy for the program using the tablebases to notice and take account of this 'feature'.

The Nalimov tablebases, which use state-of-the-art compression techniques, require 7.05 GB of hard disk space for all five-piece endings. To cover all the six-piece endings requires approximately 1.2 terabyte. It is estimated that seven-piece tablebases will require more storage capacity than will be available in the foreseeable future.

It is surprising, but easily verified, that without an endgame tablebase even otherwise very strong chess engines may fail to find a winning plan even in endings with six or fewer pieces, when they need more moves than the calculation horizon to achieve a checkmate, a win of material or the advance of a pawn. Many endings require more moves than their calculation horizon.

Other optimizations

Many other optimizations can be used to make chess-playing programs stronger. For example, transposition tables are used to record positions that have been previously evaluated, to save recalculation of them. Refutation tables record key moves that "refute" what appears to be a good move; these are typically tried first in variant positions (since a move that refutes one position is likely to refute another). Opening books aid computer programs by giving common openings that are considered good play (and good ways to counter poor openings).

Of course, faster hardware and additional processors can improve chess-playing program abilities, and some systems (such as Deep Blue) use specialized chess hardware instead of solely software implementations.

Standards

Computer chess programs usually support a number of common *de facto* standards. Nearly all of today's programs can read and write game moves as Portable Game Notation (PGN), and can read and write individual positions as Forsyth-Edwards Notation (FEN). Older chess programs often only understood long algebraic notation, but today users expect chess programs to understand standard algebraic chess notation.

Most computer chess programs are divided into an *engine* (which computes the best move given a current position) and a *user interface*. Most engines are separate programs from the user interface, and the two parts communicate to each other using a public communication protocol. The most popular protocol is the Xboard/ Winboard Communication protocol. Another open alternate chess communication protocol is the Universal Chess Interface. By dividing chess programs into these two pieces, developers can write only the user interface, or only the engine, without needing to write both parts of the program. (See also List of chess engines.)

Playing strength versus computer speed

It has been estimated that doubling the computer speed gains approximately fifty to seventy ELO points in playing strength (Levy and Newborn 1991:192).

However, this applies mainly to computer-vs-computer matches, and not to computer-vs-human matches.

Other chess software

There are several other forms of chess-related computer software, including the following:

- **Chess game viewers** allow players to view a pre-recorded game on a computer. Most chess-playing programs can be also used for this purpose, but some special-purpose software exists.
- **Chess instruction software** is designed to teach chess.
- **Chess databases** are systems which allow the searching of a large library of historical games. Scid is a good example of a chess database. Scid may be used under Microsoft Windows, UNIX, Linux and Apple's OSX used on the newer machines. There are also commercial databases, such as chessbase, although there is no commercial chess database that runs on anything other than Microsoft Windows.
- **Software for handling chess problems**

Advanced chess

Advanced Chess is a form of chess developed in 1998 by Kasparov where a human plays against another human, and both have access to computers to enhance their strength. The resulting "advanced" player was argued by Kasparov to be stronger than a human or computer alone, although this has not been proven.

Computer chess theorists

Well-known computer chess theorists include:

- D. F. Beal
- David Levy
- Robert Hyatt [1] (http://www.cis.uab.edu/info/faculty/hyatt/hyatt.html) (author of open source chess program Crafty)
- Hans Berliner
- Claude Elwood Shannon
- Vasik Rajlich (author of Rybka)

The future of computer chess?

Some observers extrapolate that computers will consistently beat the best human players by perhaps 2010, and then go on to exceed their abilities to the point where a human versus computer chess match would be as unfair as a human versus automobile race. Others are unconvinced, saying that there are still deep strategic elements to chess that will resist brute-force computer searching. In anticipation of this and inspired by the defeat of Kasparov by Deep Blue a new game called Arimaa has been invented that uses standard chess pieces and board but has different rules. The game was designed specifically to be hard for computers to play but fun for humans.

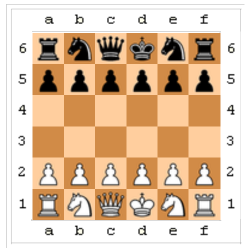
If chess computers become too big of a match for humans, the field of computer versus computer competition (where makers of chess computers pit their best machines against each other) will take over and it already has its start.

One potentially fruitful field of research is in distributed computation, where many computers are joined together through the internet and are each tasked with a small section of the overall search tree to analyse. The leading project is the ChessBrain project (http://www.chessbrain.net/) , which gained a world record in 2004 for the largest number of computers ever playing a game of chess simultaneously (2,070).

Solving chess

The prospects of completely solving chess are generally considered to be rather remote. It is widely conjectured that there is no computationally inexpensive method to solve chess even in this very weak sense, and hence the idea of solving chess in the stronger sense of obtaining a practically usable description of a strategy for perfect play for either side seems unrealistic today. However, it should be noted that neither has it been proven that no computationally cheap way of determining the best move in a chess position exists, nor has it even been proven mathematically that a traditional alpha-beta-searcher running on present-day computing hardware could not solve the initial position in an acceptable amount of time. The difficulty in proving the latter lies in the fact that, while the number of board positions that could happen in the course of a chess game is huge (on the order of 10^{40}), it is hard to rule out with mathematical certainty the possibility that the initial position allows either side to force a mate or a three-fold repetition after relatively few moves, in which case the search tree might encompass only a very small subset of the set of possible positions. Still, it can certainly be said that nothing at present indicates a practical possibility of solving chess in any sense of the word.

Chronology of computer chess



Los Alamos chess. This simplified version of chess was played in 1956 by MANIAC I computer.

- 1769, Wolfgang von Kempelen builds the Automaton Chess-Player, in what becomes one of the greatest hoaxes of its period.
- 1868, Charles Hooper presented the Ajeeb automaton — which also had a human chess player hidden inside.
- 1912, Leonardo Torres y Quevedo builds a machine that could play King and Rook versus King endgames.
- 1948, Norbert Wiener's book *Cybernetics* describes how a chess program could be developed using a depth-limited minimax search with an evaluation function.
- 1950, Claude Shannon publishes "Programming a Computer for Playing Chess", one of the first papers on the problem of computer chess.
- 1951, Alan Turing develops on paper the first program capable of playing a full game of chess.
- 1952, Dietrich Prinz develops a program that solves chess problems.
- 1956, first program to play chess-like game, Los Alamos chess (see diagram at right), was developed by Paul Stein and Mark Wells for MANIAC I.
- 1956, invention of alpha-beta search algorithm by John McCarthy
- 1958, NSS becomes the first chess program to use alpha-beta searching.
- 1958, first programs that could play a full game of chess were developed, one by Alex Bernstein and one by Russian programmers using a BESM.
- 1962, first program to play credibly, Kotok-McCarthy published at MIT
- 1966-1967, first chess match between computer programs. Moscow Institute for Theoretical and Experimental Physics (ITEP) defeated Kotok-McCarthy at Stanford University by telegraph over nine months.
- 1967, Mac Hack Six, by Richard Greenblatt et al. introduces transposition tables and becomes the first program to defeat a person in tournament play
- 1970, first year of the ACM North American Computer Chess Championships
- 1974, Kaissa wins first World Computer Chess Championship
- 1977, the first microcomputer chess playing machine, CHESS CHALLENGER, was created
- 1977, establishment of the International Computer Chess Association
- 1980, Chess 4.6 becomes the first chess computer to be successful at a major chess tournament.
- 1980, first year of the World Microcomputer Chess Championship
- 1980, establishment of the Fredkin Prize.
- 1981 Cray Blitz won the Mississippi State Championship with a perfect 5-0 score and a performance rating of 2258. In round 4 it defeated Joe Sentef (2262) to become the first computer to beat a master in tournament play and the first computer to gain a master rating.
- 1982, Ken Thompson's hardware chess player Belle earns a US master title.
- 1988, HiTech developed by Hans Berliner and Carl Ebeling wins a match against grandmaster Arnold Denker 3.5 - 0.5.
- 1988, Deep Thought shares first place with Tony Miles in the Software Toolworks Championship, ahead of a former world champion Mikhail Tal and several grandmasters including Samuel Reshevsky, Walter Browne, Ernst Gruenfeld and Mikhail Gurevich. It also defeats grandmaster Bent Larsen, making it the first computer to beat a GM in a tournament. Its rating for performance in this tournament of 2745 (USCF scale) was the highest obtained by a computer player.
- 1989, Deep Thought loses two exhibition games to Garry Kasparov, the reigning world champion.
- 1992, first time a microcomputer, the Chessmachine Gideon 3.1 by Ed Schroeder, wins the 7th World Computer Chess Championship in front of mainframes, supercomputers and special hardware.
- 1997, Deep Blue wins a six-game match against Garry Kasparov +2-1=3 (see [2] (http://chess.ibm.com)).
- 2002, Vladimir Kramnik draws an eight-game match against Deep Fritz.
- 2003, Kasparov draws a six-game match against Deep Junior.
- 2003, Kasparov draws a four-game match against X3D Fritz.
- 2005, Hydra defeats Michael Adams 5.5-0.5.
- 2005, a team of computers (Hydra, Deep Junior and Fritz), wins 8.5-3.5 against a rather strong human team formed by Veselin Topalov, Ruslan Ponomariov and Sergey Karjakin, who had an average ELO rating of 2681.
- 2006, The undisputed world champion, Vladimir Kramnik, is defeated 4-2 by Deep Fritz. One of the defeats is caused by Kramnik missing the computer mate-in-one.

See also

- Advanced Chess
- Arimaa
- Chess Engines Grand Tournament
- Chess engines
- Chess960
- Computer Go
- Computer Olympiad
- Swedish Chess Computer Association
- World Computer Chess Championship

Notes

1. ^ http://www.rebel.nl/anand.htm
2. ^ http://www.chessbase.com/newsdetail.asp?newsid=2476
3. ^ http://kd.lab.nig.ac.jp/chess/tablebases-online/
4. ^ http://www.xs4all.nl/~timkr/chess2/diary_16.htm

References

- Levy, David & Newborn, Monty (1991), *How Computers Play Chess*, Computer Science Press, ISBN 0-7167-8121-2

- Nunn, John (2002), *Secrets of Pawnless Endings*, Gambit Publications, ISBN 1-901983-65-X

- Shannon, Claude E. (1950), "Programming a Computer for Playing Chess" (http://archive.computerhistory.org/projects/chess/related_materials/text/2-0%20and%202-1.Programming_a_computer_for_playing_chess.shannon/2-0%20and%202-1.Programming , *Philosophical Magazine*, vol. Ser.7, Vol. 41, no. 314

External links

- Mastering the Game: A History of Computer Chess (http://www.computerhistory.org/chess/) at Computer History Museum (http://www.computerhistory.org/)
- The History of Computer Chess: An AI Perspective. Watch Full Lecture - WMV 183MB (http://archive.computerhistory.org/lectures/history_of_computer_chess_an_ai_perspective.lecture.2005-09-08.102656913.wmv) | Google Video (http://video.google.com/videoplay?docid=-1583888480148765375) featuring Murray Campbell (IBM Deep Blue Project), Edward Feigenbaum, David Levy, John McCarthy, and Monty Newborn. at Computer History Museum (http://www.computerhistory.org/)
- History of computer chess (http://www.chessbase.com/columns/column.asp?pid=102)
- Bill Wall's Computer Chess History Timeline (http://www.geocities.com/SiliconValley/Lab/7378/comphis.htm)
- The History of Computer Chess Video (http://www.bobby-fischer.net/bobby_fischer_video_18.htm) (2 hour video - series of lectures)
- ChessComputers.org (http://www.chesscomputers.org) Site dedicated to electronic chess machines or dedicated chess computers
- Java Chess Game (http://www.supreme-chess.com/java-chess-game/java-chess-game.html) - Play a game of chess against the computer.
- Free Chess Software (http://freechess.50webs.com/index.html) - Some free chess software
- ClassicChessAndGames.com (http://www.classicchessandgames.com) Electronic chess computers sets from Excalibur, Fidelity, Mephisto, Saitek, and other electronic board games.
- Chess Software at Chess Books Online (http://www.chess-books-online.com/chess-software/)
- Scid (http://scid.sourceforge.net/) - chess database (this is free to download and use)
- Defending Humanity's Honor (http://www.xs4all.nl/~timkr/chess2/honor.htm) , an article by Tim Krabbé about "anti-computer style" chess
- Guardian article about the state of computer chess in 2002 (http://www.guardian.co.uk/online/story/0,3605,817484,00.html)
- Computer-Chess Club (http://www.talkchess.com/) - Bulletin board where professional authors discuss their programs
- WBEC Ridderkerk: Information on a large collection of free chess engines and inter-engine tournaments (http://wbec-ridderkerk.nl/)
  - WBEC Ridderkerk: Chess programming links (http://wbec-ridderkerk.nl/html/ProgrLinks.html)
- A guide to Endgame Tablebases (http://horizonchess.com/FAQ/Winboard/egtb.html)
- Omid David Tabibi's Computer Chess Publications (http://www.cs.biu.ac.il/~davoudo/pubs.html)
- GameDev.net -- Chess Programming by François-Dominic Laramée (http://www.gamedev.net/reference/programming/features/chess1/)
- Colin Frayn's Computer Chess Theory Page (http://www.frayn.net/beowulf/theory.html)
- "How REBEL Plays Chess" by Ed Schröder (http://members.home.nl/matador/Inside%20Rebel.pdf) (PDF)
- "An Enjoyable Game": How HAL Plays Chess (http://mitpress.mit.edu/e-books/Hal/chap5/five1.html) eBook chapter about Poole - HAL 9000 and computer chess in general
- "Play chess with God" (http://plan9.bell-labs.com/~ken/chesseg.html) - Play chess against Ken Thompson's endgame database
- GE Chess: (http://www.gechess.org/) A multiplayer 3D Google Earth Chess Free Games Network
- You Tube movie (http://video.google.com/videoplay?docid=-1583888480148765375&q=chess&hl=en)

Categories: Articles with unsourced statements | Computer chess

---