What computers teaches us about chess

Fred Senekal
6 March 2019

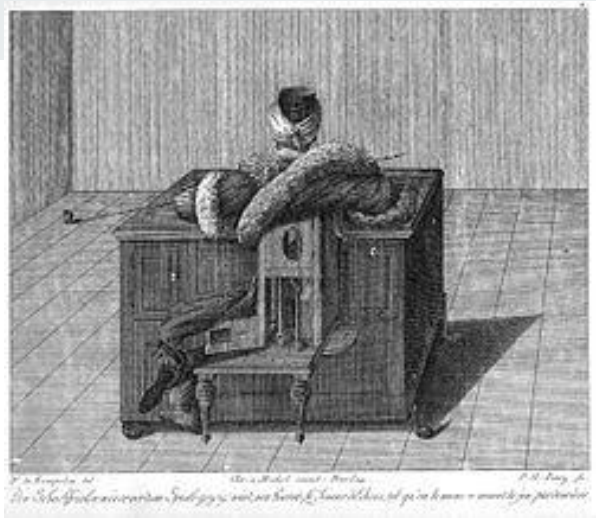
Overview

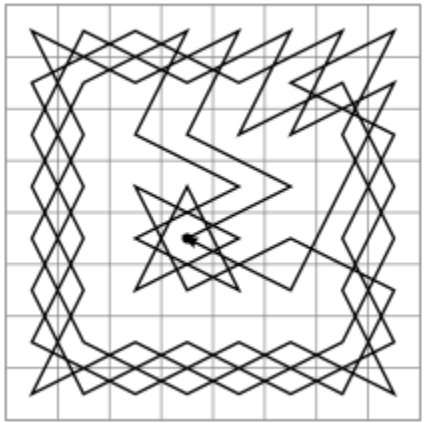
- History
- Current chess engines
- Evaluation function
- Building a search tree

History

1769

- Wolfgang von Kempelen builds the The Turk (Automaton Chess Player)
- Gave the illusion that it was an intelligent playing machine, but was in fact operated by a skilled operator.
- Won many games in Europa and the Americas, including games against Napoleon Bonaparte and Benjamin Franklin.





1950's

- 1950 - Claude Shannon publishes "Programming a Computer for Playing Chess" - one of the first papers on the problem of computer chess.
- 1951 - Alan Turing publish the first program, developed on paper, capable of playing a full game of chess. Known as Turochamp.
- 1956 - Los Alamos chess is the first "chess-like" program, playing on a MANIAC I computer
- 1956 - John McCarthy develops the alpha-beta search algorithm.
- 1957 - First programs to play a complete game of chess developed by Alex Bernstein.

1960's

- 1958 - First program to play credibly, Kotok-McCarthy published by MIT
- 1963 - Grandmaster David Bronstein defeats M-20 running an early chess program.
- 1966-67 - First chess match between computers. Moscow Institute for Theoretical and Experimental Physics defeats Kotok-McCarthy at Stanford University by telegraph over nine months.

1970's

- 1974 - The first World Computer Chess Championship is won by Russian program Kaissa.
- 1975 - First commercial microcomputer, the Altair 8800.
- 1976 - First dedicated commercial chess computer, Chess Challenger released by Fidelity Electronics.
- 1976 - Microchess is the first commercial software for a microcomputer.

1980's

- 1981 - Cray Blitz wins Mississippi State Championship with a perfect score of 5-0. First computer to defeat a master.
- 1981 - IBM releases the PC.
- 1986 - Chessmaster 2000 released - to become the world's best selling line of chess programs.
- 1987 - Chessbase founded by Frederic Friedel and Matthias Wullenweber - the first chess database program.
- 1988 - Deep Thought becomes the first computer to beat a GM in a tournament, performance of 2745
- 1989 - Deep Thought loses two exhibition matches against Gary Kasparov.


ACM Chess Challenge
Garry Kasparov
vs

1990's

1992 - ChessMachine wins the World Computer Chess Championship. It is the first time a microcomputer beats a mainframe.

1993 - Deep Blue loses a four game match against Bent Larsen

1996 - Deep Blue loses a six-game match against Garry Kasparov.

1997 - Deep Blue wins a six-game match againt Garry Kasparov.

2000's

- Early 2000's - Various matches between computers and humans: Kramnik vs Deep Fritz, Kasparov vs Deep Junior, Kasparov vs X3D Fritz, Hydra vs Adams, etc.
- 2005 - Rybka wins IPCCC tournament and quickly becomes the strongest engine.
- 2008 - Stockfish 1.0 released in November 2008

2010's

- 2014-2019. Stockfish won various computer world chess championships.
- 2019. Stockfish 10 is the world's strongest publicly available chess engine - it is free and open source.
- 2017 - AlphaZero from Google beats Stockfish 28-0, with 72 draw in a 100 game match.
- 2017 - AlphaZero taught itself to play chess using 5000 TPUs using neural networks. After 4 hours, it could play stronger than Stockfish 8 and after 9 hours of training, beat Stockfish in a controlled 100-game tournamentt.
- 2018 - AlphaZero paper published in journal Science on 7 December 2018.

How strong are chess computers?

- Current estimates for the strength of the best chess engines are at about 3300 to 3500.
- Magnus Carlsen currently has a rating of 2845.
- This roughly means that the best humans could score about 2 to 4 games in a 100 game match.

“Playing against the computer is like running against a Ferrari”

The beginning and the end...

Opening Books

- All chess engines have extensive opening books from which it makes its first moves.
- In each position, there are a number of moves that can be played.
- Each move has an associated probability - the engine makes a move based on these probabilities.
- Once the moves in the opening book is depleted, it starts making moves based on evaluation.

LESSON:

- You simply NEED TO KNOW your opening lines to play them accurately like a computer.

Endgame Tablebases

- All positions up to 7 pieces (including 2 kings) have been completely analysed and can be perfectly played.
- Research to generate all 8 piece positions are currently underway.
- Once the computer reaches a 7 piece endgame (or fewer pieces in many cases), it consults an endgame tablebase and plays a perfect game.
- The endgame tablebases has opened new understanding of endgame theory.

LESSON:

- You need to STUDY the various techniques for correct endgame play and become skilled in various material setups.

Forced mate in 546… white to move

Number of pieces	Number of positions	Database name	size
2	462	Syzygy	(included in the 5-piece tablebase)
3	368,079	Syzygy	(included in the 5-piece tablebase)
4	125,246,598	Syzygy	(included in the 5-piece tablebase)
5	25,912,594,054	Syzygy	939 MB
6	3,787,154,440,416	Syzygy	150.2 GB
7	423,836,835,667,331	Syzygy	17 TB
7	423,836,835,667,331	Lomonosov	140 TB
8	38,176,306,877,748,245	n/a	1 PB

Evaluation Function

What is an evaluation function?

- Value that expresses the strength of a position.
- Usually values larger than 0.0 indicate a superior position for white and values smaller than 0.0 indicate a superior position for black.
- Usually normalised such that the value for a pawn is about 1.0.
- A computer needs to compute an evaluation for each position it considers - it precisely expresses the evaluation of the position.
- The goal would be to optimize the evaluation function for the player over time.
- Usually, white starts with an advantage of about 0.1 to 0.2
- 0.5 - slight advantage, >1 - advantage, >3 - decisive advantage

Elements that go into the evaluation function

- Piece value
- Piece square bonuses
- Material imbalance
- Pawn structure (isolated, backward, doubled, connected, connected bonus)
- Piece bonus (penalties to pieces of given colour and type - knight outposts, bishop outposts, trapped rooks, long diagonal bishop, weak queen, etc.)
- Mobility
- Threats (hanging, overload, king threat, pawn push threat, rank threat, weak unopposed pawn, etc.)
- Passed pawn
- Space
- King attack (king danger, shelter strength, pawnless flank, etc)
- Initiative

Lessons from the evaluation function

- You need to develop the ability to ACCURATELY access the position.
- If you do not consider ALL aspects of the position, you may completely miss some critical contributions to the evaluation of the position.

Building a search tree

Creating a search tree

- A computer needs to construct a search tree of the possible moves from the current position.
- It uses the evaluation function to determine a score associated with each of the generated positions.
- It chooses its moves based on a maximization of the evaluation function.

Branching Factor

- How many moves can you make in any given position?
- In the beginning, there are 20.
- The average over the duration of a chess game, is about 35.
- This is known as the branching factor.
- After two “plies” (a move by you and your opponent), there can be on average 35 x 35 = 1225 possible positions.
- After only 5 moves by you and your opponent, there are 35 ^ 10 = 2758547353515625 possible positions...

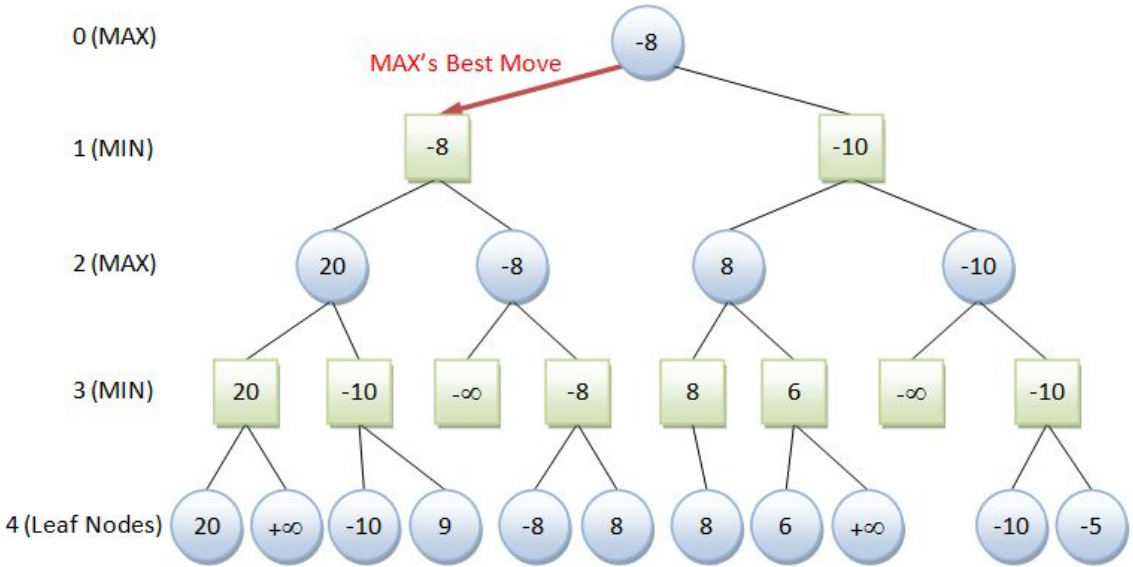
Reducing the branching factor

- It is impossible to consider all the possible variations.
- To reduce complexity, only a few branches (2 to 4) are considered at each position.
- The computer uses heuristics (possible best move first) to determine which of these to consider first.
- Practice has shown that it is better to analyse deeper, but fewer moves at each point.

LESSON:

- You need to TAKE YOUR TIME to at least consider some continuations for every position.
- You need to IDENTIFY CANDIDATE MOVES for further consideration.

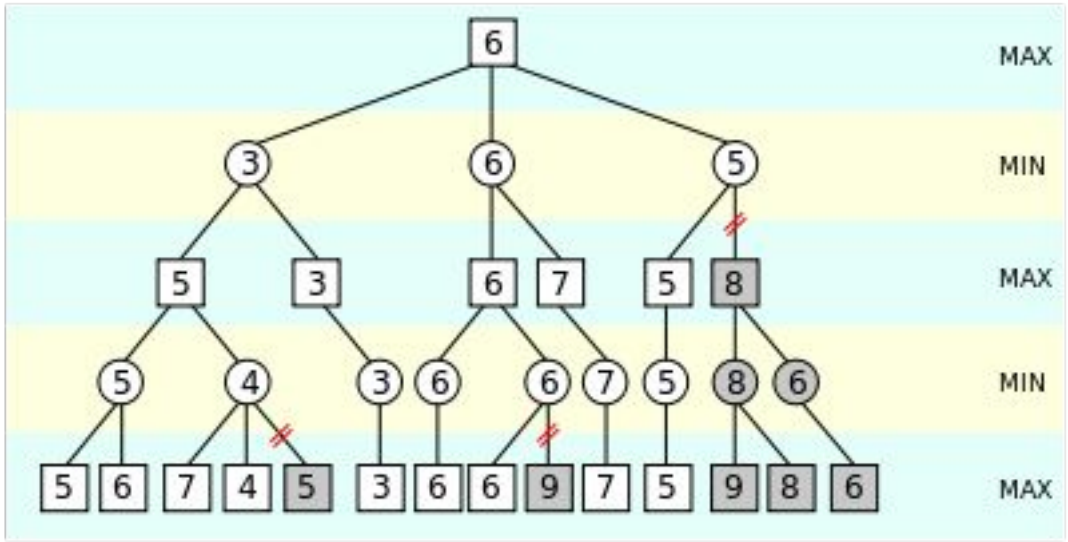
The MiniMax Algorithm



Lessons from the MiniMax algorithm

- You should always ASSUME that your opponent will play at their BEST.
- Choose moves that are GUARANTEED to give your position the best possible evaluation, not moves where your opponent has potential to optimize their evaluation.
- Consider each POSITION ONCE.

Alpha Beta Pruning



Lessons from Alpha-Beta pruning

- Once you find a REFUTATION of one of the moves you consider, you should immediately stop the evaluation of the line, and backtrack the line to a prior position.
- Attempt to CUT OUT unnecessary thinking as soon as possible.

The NULL move heuristic

- A good heuristic for when the evaluation of a position does not significantly change, is the NULL move heuristic.
- If a player can forfeit a move and his position does not significantly weaken (the evaluation of the position does not change much), then the line need not be further considered.

LESSONS:

- Sometimes it is worth checking the strength of your position by considering what would happen if your opponent could make two moves consecutively.
- Sometimes you need to stop to consider further moves when your candidate move is good enough.

Thank You