Introduction

IA in Chess is quite complex

• Even the fastest computer cannot solve the chess game (cannot analyze every possible situation)

• There are many ways of reducing the need of computational power

The best machine is now slightly better than the best human player
History

1769 – The Turk

Automaton made by Wolfgang van Kempelen
Played famous people including Napoleon
Obviously cheated (no-legged man inside)

Introduction
History
Glossary
The basis
Search techniques
Position Evaluation
Use of databases
Discoveries thanks to Computer Chess
Future of IA in chess (and more)

1912 – El Ajedrecista

Automaton built by Leonardo Torres y Quevedo
Played an endgame with King and Rook vs King
Could always win
Could detect illegal moves
**History**

**1950s** – Very beginning of computer chess

**1960s** – Good enough to beat novice players sometimes

**1977** – First tournament win for a program (*Chess 4.5, Minnesota Open*)

---

**Computers vs World Champions**

**1985** – Kasparov vs 15 chess computers. Won every single game

**1992** – Kasparov vs Fritz 2 in a speed game. Won 6.5 – 4.5. First time a computer defeats a world champion

**1996** – Kasparov vs Deep Blue (IBM), won 4-2. First time a computer defeats a world champion under tournament conditions

**1997** – Rematch: Kasparov vs Deep Blue II, lost 2.5 – 3.5. A great milestone in computer chess (…)

**2006** – Kramnik vs Deep Fritz, lost 4-2
**Glossary**

- **Pawn**
- **Knight**
- **Bishop**
- **Rook**
- **Queen**
- **King**

_A ply:_ one move for one player

_A move:_ two plies

---

**The basis**

It’s impractical to consider every possibility so choices must be made by the use of _search techniques_.

A chess computer...

- Usually builds the whole tree for the few next plies.
- Decides which nodes are worth a research.
- Uses MANY features to decide how deep the search of a node should be.
Search techniques

The Minimax algorithm

A value is affected to each « final » position

The current player will choose the path that gives him the best moves and gives the opponent the worst ones

Problem : Too many positions in chess
- Average of 30 legal moves
- Typical games take 80 plies

Number of positions to evaluate: $30^{80} = 1 \times 10^{118}$

Note: It is believed that there are $10^{75}$ atoms in the Universe.

Alexandre Gimenez - Artificial Intelligence in Chess

Search techniques

Alpha-beta pruning

Improvement of the Minimax algorithm that reduces the number of nodes that must be evaluated

Stops evaluating a move (triggers a cutoff) when at least one better possibility has been found elsewhere

Alpha-beta pruning usually allows to double the efficiency of the minimax algorithm (when using killer heuristic). It makes the search non-uniform, which is obviously very important.

Alexandre Gimenez - Artificial Intelligence in Chess
Search techniques

Credits

Credits = certain amount of extra plies. Can be negative.

The algorithm evaluates a 10 plies depth tree. The credit generation will allow some moves to be evaluated deeper and will prevent some other moves from reaching the 10th ply.

In which cases a move is given credit:

- Forced moves: there are less legal moves so the subtree is smaller than usual (i.e. faster to analyze)
- Principal variation: in the tree, one move seems to be very good for both players
- Singularity, binarity,... (a move is much better than the others)

http://turbulence.org/spotlight/thinking/

Alexandre Gimenez - Artificial Intelligence in Chess
We could see that the whole algorithm depends on evaluating how good a position is (what is its score)...

But it's actually very complex to tell how good a position is. Many factors, with different weight, that varies with time, must be considered.

The estimated value of a position is:

\[ V = \text{factor1} \times \text{coeff1} + \text{f2} \times \text{c2} + \text{f3} \times \text{c3} + \ldots \]

The coefficients are parameters that can change depending on the time of the game (beginning, middle or end), the opponent... and we don't know their optimal values.

List of factors

Material value
- Pawn 1, Knight 3, Bishop 3, Rook 5, Queen 9, King infinite. This is a simplified approach... click here

King safety (stay in the corner at the beginning, have a pawn shield,...)

Mobility (knights rather in the center)

Pawn structure (isolated pawns, blocked pawns,...)

Was the move ever played in a top-level game?
Use of databases

Databases of pre-processed position evaluation and of hundreds of famous games from top players are used in some cases: mainly for openings and endgames.

Openings

Database of early possible opening positions

Deep-blue: Database of 4,000 positions that have been processed. There is a statical evaluation value for those 4,000 positions. This way, Deep Blue can choose positions that will provoke a type of game that it often wins.

Endgames

The less material remains on the board, the less legal moves there are, the smaller the tree is... It becomes possible to compute every single position.

This has been done for every positions with 5 pieces or less on the board. The results are stored in the endgame database (7Gb). Additionnaly, some 6 pieces positions are also stored. Saving all 6 pieces positions would take 1.2Tb.
Discoveries

A few discoveries have been made by chess computers, especially for endgames (when the full databases are available, so they can be played perfectly).

Fifty move rule: « The fifty move rule states that a player can claim a draw if no capture has been made and no pawn has been moved in the last fifty consecutive moves ». Positions* believed to be won turned out be drawable if played correctly. The fifty moves rule has even been (temporarily) modified because of this.

But it actually does not make any difference for the human player, who doesn’t manage to play the position perfectly.

*• Queen vs 2 bishops • Queen vs 2 knights • 2 bishops vs a knight • 2 knights vs a pawn • Rook and bishop vs rook

Future

Future of Chess

- System that would always beat a human (should happen quite soon)

- Interesting project: Chess Brain
  Uses a grid (thousands of computers connected through Internet) in order to use a much higher computational potential

- Solving chess? (very unlikely)
Future of AI in board games: The Go game

- Far more complex than chess (larger board – 19*19 - much more legal movements for each ply, the game gets more and more complex).

- The AI system probably needs to understand how the opponent thinks and adapt to his gamestyle.

And for those who want to get rich: Poker! The current systems are very efficient.