Deep Blue

Christoph Schulze

Introduction

Computer Chess has been a popular problem in Computer Science starting with Claude Shannon's paper: *"Programming a computer for playing chess"*[1] that was published in 1950. In this ground laying paper he already proposed ideas how a chess computer could work and some of his ideas are still being used in today's chess programs[1].

One of the most famous duels between man and chess machine was that of Gary Kasparov, the highest ranked chess players of all time[2] and Deep Blue, an IBM chess supercomputer. While Kasparov was able to beat the predecessors Deep Thought (2-0, 1991[3]) and Deep Blue I (4-2, 1996[4]) he eventually lost against Deep Blue II (3½–2½, 1997[4]). This was the first time that a machine could beat the reigning chess world champion. IBM retired Deep Blue II after the event and they never took up the challenge of Kasparov for a rematch.

Kasparov claimed that IBM cheated during game 2, accusing IBM technicians of interfering during the game. IBM denied these claims and argued that they behaved within the rules of the tournament and only worked on the computer in between matches to adjust it using information from the game that was played before. They also released the log files[5] of the matches, which proof that there was no human interference during the game, although that happened with a very long time delay.

What required a supercomputer 15 years ago is now available for nearly everybody. Today's chess software in conjunction with cheap powerful consumer hardware gives you chess at a grand master level[6] in your own home. Houdini is an example of such a program. It is currently the highest rated[7] chess engine available. The ranking is established by letting the different chess systems play against each other on a standardized hardware platform to gauge the quality of the engine alone and factor out the computer hardware.

The Hardware of Deep Blue II

Deep Blue[2] is a massively parallel computer that was solely designed to play chess. It features 30 processor nodes, where each node contains on multi-purpose CPU and 16 chess chips, which are specifically designed to do chess search operations. One central node is used as a command node that does the top level evaluation of the search tree and divides the deeper workloads onto the other 29

---

[1] Albeit with a tremendous amount of modifications and extensions
[2] http://en.wikipedia.org/wiki/Comparison_of_top_chess_players_throughout_history
[3] http://www.nytimes.com/1990/01/14/magazine/kasparov-beats-deep-thought.html?pagewanted=all&src=pm
[4] http://www.worldchesschampions.com/kasparov_vs_deepblue.php
[5] http://researcher.watson.ibm.com/researcher/view_project.php?id=2942
[6] http://en.wikipedia.org/wiki/Chess_Grand_Master
[7] http://www.chess.com/article/view/records-in-chess

nodes. It can evaluate up to 300 million positions per second (the actual number depends on the game layout it varies between 100-300 million)

How Deep Blue II Plays Chess

Deep Blue [2] follows the following three steps in order to determine a move:

- Search the space of all possible moves and future position. This is usually a search operation on a tree structure. However, since the search space is extremely large (10^120 possible positions in an average chess game of 40 moves) a brute force approach is infeasible and several optimizations have to be used.
- Evaluate the positions and rank them
- Decide which move to use, by using the ranked search results and moves from playbooks

Deep Blue shares some similarities to how Grandmaster's play chess, but it also has some key differences. It usually starts the game with a set of moves from its opening book, a database with standard openings that were developed by other grand masters and which Deep Blue has in its memory. The openings were optimized for games that would favor Deep Blue's style of play. They also included some specific openings against Kasparov, but they were apparently not used in any of the matches. (The opening was randomly chosen from a predefined subset of all openings)

In addition to the opening book it has two other databases. The extended book and the endgame database are used to guide its moves during mid and endgame. The endgame database contains special scenario for plays with six or less pieces.

However, most of the time Deep Blue works independent of those databases and just uses them as additional input in its decision making process. (E.g. the endgame database was only used in one of the six games) In order to determine the next move Deep Blue utilizes a massively parallel search engine that tries to look several moves ahead. Since the search trees that it operates on are very broad and can be tremendously deep, it uses several optimizations to prune the search tree. This allows it to search deeper. Deep Blue is able to search branches as deep as 40 moves and has an average of 12.2 moves over all possible branches. Searching all possible branches is important for Deep Blue to be able to guard against simple mistakes that could occur if certain paths are not evaluated deep enough.

Figure 1 Rook on seventh rank

Human players on the other hand can rely on intuition and often immediately know which moves are bad and can therefore prune more efficiently. Deep Blue does not have a notion of intuition it needs to evaluate the possible moves until it reaches a point where it decides that it is a path not worth taking or that ends in a known position.

The search utilizes a credit system to judge the value of a path. (E.g. a move that leads into a threat or mate threat situation) These values are used to decide whether to keep searching a branch or not. An evaluation function is employed to determine the value of a given position.

The evaluation function was improved after Deep Blue I's matches against Kasparov. (The other changes were mostly increases in compute power and tweaks to increase the search speed). Deep Blue's original evaluation function had about 6400 features. After analyzing the first round of games against Kasparov and sparing matches against other grand masters they were increased to around 8000. Its features were mostly hand tuned and represent certain characteristics of a chess game. (E.g. rook on the seventh rank[8], see Figure 1).

How Humans Play Chess

Chess grandmasters rely heavily on a very good visual memory to recognize patterns in the game and then use a predefined tactics for the given situation. This way they don't have to calculate too many moves in advance. But even so thinking ahead is also one of the key strengths that a good chess player needs to have. Depending on the situation a human can think ahead up to fifteen moves, however most of the time this number is in the lower end between 3-5 moves[9]. Pruning is very important here, good chess players have to have a good intuition for bad moves, moves that they don't even have to calculate anymore since they immediately know that they won't help them. [10]

Several tactics were developed to specifically target weaknesses of computer chess systems and the way the evaluate positions. These often involve very long term goals that the human player tries to gain that lie beyond the evaluation limit of the computer or using specific opening tactics that force the computer into positions were it would be at a disadvantage. (E.g. moves that are not covered in the opening book)

Other AI Game Challenges: Checkers and Go

Chess is probably one of the most known board games played by computers but it is not the only one. Checkers is an example of a game that is considered solved, i.e. there exists a solution that cannot be beaten and if played without an error always ends in a draw[11]. Go on the other hand is the example of a problem that is even harder than chess. For one the game board is much larger than that of chess and which makes evaluation of the search tree much harder. Furthermore the game gets more complex as it progresses (more pieces are added that offer more possibilities) whereas it gets less complex in chess where pieces are removed from the board during the game.

References

[1] C. Shannon, "Programming a computer for playing chess," *Philosophical Magazine*, vol. 7th series, no. no. 314, pp. 256–75, 1950.

[2] M. Campbell, A. J. Hoane, and F. Hsu, "Deep Blue," *Artificial Intelligence*, vol. 134, no. 1–2, pp. 57–83, Jan. 2002.

---

[8] http://www.chess.com/article/view/the-power-of-two-rooks-on-the-

[9] http://www.maa.org/mathland/mathland1.html

[10] http://www.youtube.com/watch?v=NJarxpYyoFI

[11] http://www.nytimes.com/2007/07/19/science/19cnd-checkers.html?hp&_r=0