Bogdan Costea

AIND Feb Cohort

Deep Blue (https://pdfs.semanticscholar.org/ad2c/1efffcd7c3b7106e507396bdaa5fe00fa597.pdf)

The purpose of the paper is to describe the Deep Blue chess system: brief history, search algorithms, evaluation functions, constraints.

Deep blue is a chess system that managed to beat in 1997 the chess world champion: Garry Kasparov. The system is the culmination of multiple attempts for creating a performant chess machine and was composed both from software and hardware implemented algorithms. A previous version of the machine, dated 1 year earlier, failed to achieve the expected result but a series of improvements allowed the second version to score higher against Garry Kasparov. First improvement was the redesign of the chess chips, basically hardware search engines, allowing them to accept more evaluation features, features that would respond to problems observed during the first round against the World Champion.

Deep Blue is organized on three layers: on top, there is a PS processor which acts as a master, controlling the remainder processors delegated at the status of workers. There is almost no parallelism at the master level as it searches the top level of the game tree and then distributes its to the workers. The leaves of each of the worker's subtrees are then distributed after a brief search to the hardware chess chips which executes the search deeper in the game tree. The paper abounds in performance numbers for each level of the tree, numbers which I don't think are relevant to reproduce here. It is relevant though to mention that the system would have different performance for more tactical positions as opposed to quitter position.

The algorithms employed by the designers of Deep Blue were built upon quiescence search, iterative deepening, transposition tables, negascout (principal variation search). Also, it benefits from a consistent opening and end game databases. Specific to Deep Blue is though the fact that the search is highly ununiform. These derives from the fact that strong chess players are able to predict games situations deeper than a computer simply by ignoring weak game situation and focusing on powerful and creative moves. Another specific is the fact that the evaluating functions were hardware implemented which would simplify the task of writing the software but would impede adding new evaluation features. Of particular importance is the fact that no matter how complex the function is would execute in constant time allowing the search algorithm to go to the same depth as a fast-superficial function but with superior evaluations of positions. Also specific is the mixed hardware – software approach to search allowing for a high degree of parallelism.

The hardware search is based on the chess chip. They generate the moves and orders them per evaluation functions. The evaluation functions can be "slow" or "fast" depending on the criteria used. For example, fast would be assessing piece capturing, piece placement value. Slow would be positional advantage, pawn structure, trapped pieces etc.

The software search is designed to deepen the search for interesting situations and the paper describes in detail the forcing/forced(FF) moves. Basically, the deepening of the search tree is prepared by accumulating a quantifiable criterion which at a certain threshold would trigger a deeper search. This allows to avoid situations in which a FF flurry would end up in a meaningless position while also being

able to allocate precious resources for the situations which would bring the machine in a great advantage/win situation or allow defeat avoidance.

Deep Blue also used a substantial Opening Book database whose preparation was facilitated by Grandmasters. The positions within the book were chosen based on the fact that the machine could develop well. Also, attention was given to the particular opponent that Deep Blue had to face. There was also an extended book database which would allow the computer to find an optimal play by giving high scores to moves preferred by Grandmasters. This, even though not dictating directly a position, would prune more branches allowing for a better search.

There is also an end-game database which included all possible positions with 6 pieces left on the table.

Lastly the time management for the system is explained: the machine would take for a move an amount of time equal to the remaining time divided by the remaining moves. If a special type of conditions would occur the time for a move can be prolonged to up to third of the remaining time.