Paper review

Deep Blue (by Murray Campbell, A. Joseph Hoane Jr. b, Feng-hsiung Hsu c)

Reviewed paper starts with the development history of the Deep Blue computer predecessors and hardware (later as HW) setup. This seems interesting, but not from the point of AI software engineering techniques as the history is based on simple hardware setup and basic techniques overview.

As it is stated in the paper Deep Blue is composed of 30 nodes (processors) and 480 single-chip chess search engines. As it is stated later the whole search engine is separated to software (later as SW) and HW part, where the HW part can be highly tuned and is faster, but suffers from loss of flexibility (hard to change on demand) and the SW part written in C being extremely flexible.

The complete decision making process can be then separated to these parts:

1. Search itself - both HW and SW searches are based on Alpha-Beta search tree algorithm, where in case of HW search it implemented with null-window search and for SW search implemented with depth limitation
2. Evaluation function - according to the paper there have been over 8000 patterns used for evaluation. These patterns are connected to chess rules, but it is quite interesting that the rules are so complex that they are evaluated only at the root of the search tree.
3. Move generator - an 8 × 8 array of combinatorial logic able to generate up to 700,000 positions per second. It had additional functionality such as creating checking and check evasion moves, special attack moves and even permitting quiescence search
4. Opening and extended book - Opening book is basically a fine tuned database of openings created by chess grandmaster containing scores for each position based on human evaluation and Deep Blue testing matches, extended book is huge database of game positions with evaluation function taking to consideration number of times the move was used or even how skilled was the player who used it.

Core technique introduced in software search is Alpha-Beta pruning and either with or without depth limited search. Depth limited search mean that the algorithm keep track of tree depth it searched through and stop when selected depth is reached, this is meant to reduce computation time when there is limited time to reach best move.

Another interesting techniques is null-window which is used to extend Alpha-Beta in a way that we assume that the currently searched tree is superior to the ones that were not searched yet until proven otherwise. Implementation consists of searching through first sub-tree with usual Alpha-Beta([alpha,beta]), the rest is searched with limited window([alpha,alpha+1]), where alpha is the best Minimax value find so far. If the value returned is less or equal to alpha the assumed superior tree is indeed the best so far, otherwise the newly found tree have to be researched to get the MInimax value. The advantage is that is easier to test inferiority than calculate the exact minimax value which results in better performance.

Other interesting technique was used for finding features in evaluation functions that were “noisy” - noisy are the one that do not change much even when the weights used in the change quite a lot and to find this function the hill-climbing approach was used to explore selected features and those that did not converge were candidates for further hand examination.

At the end the Deep Blue achieved intended goal to develop world-class chess machine as it won over at that then-reigning World Chess Champion Garry Kasparov. The conclusion presented on in paper stated that despite achieving their goal, most of the areas could have been made more complete or better optimized.

Link to original paper:
https://pdfs.semanticscholar.org/ad2c/1efffcd7c3b7106e507396bdaa5fe00fa597.pdf