Deep blue summary

Deep Blue paper introduces several concepts which are briefly described below.

Transposition tables

Transposition tables are used in games such as chess where a player has all the information of the game. As the search tree can be too large in some occasions and as a single position can be achieved by different player's moves, transposition tables are implemented. A snapshot of the board or a hash to uniquely identify the board is stored in a table along with the corresponding score. Before calculating the score of a given move, this table is searched for the same hash. When found, search is avoided and the same score is returned. Otherwise, the new calculated score is stored in the table.

Extended book

A collection of "good moves" is stored. When the same board is presented, this move is favoured. The collection is made up of grandmasters moves made in previous games. This requires memory usage and a certain score to be given to each move which in Deep Blue's case was calculated by the level of each grandmaster. This reduces search but requires big amounts of information prior to starting a game.

Endgame databases

Endgame databases are somehow similar to opening book moves. There is a database containing the game result given a certain board. If one of those boards is achieved during game, the final result is assumed to be the same, without need to do any further search. A decision is taken depending on the final result of the stored game (if it is beneficial or not for the player).

Software search

Instead of using a generic search strategy, a more specific search called *dual credit with delayed extensions* was implemented in software. It was based in several principles such as *forced pairs*: It is common to force the other player's moves to delay a certain defeat or it might be necessary to make some sacrifices for a higher return in the future. This could cause search taking too much so techniques are needed to identify these situations and foresee the result of those moves in order to avoid the otherwise necessary search. (This principle could be applied to Isolation, for instance trying to force or avoid partitions).
Other principles explained in the paper are not covered in this summary because of the limited extension.

Hardware search

Hardware is faster than software to perform a given task although it is not as flexible. Using specialized hardware to perform small, atomic tasks and controlling it by software, search speed was increased.

Parallel search

Search time is one of the biggest constraints faced implementing Deep Blue. Using parallel search instead of a serial search improved Deep's Blue performance. While this is achieved today with multiple processors available in almost any notebook, in 1997 this was a significant improvement. To use this efficiently and avoid searching the same leafs by two different workers, a hierarchy was needed.

Conclusion

Not a single strategy is responsible for Deep's Blue beating Kasparov. The sum of many strategies applied together took the hardware and software capabilities to the limit and resulted in a victory. The Kasparov match was won because of the use of the described features as well as others that could not be covered such as credit generation and variable time control, applied together with a very specific heuristic that analyzed hundreds of variables such as pawns positions, blocked pieces, etc.

Rodrigo Beceiro
March 2017