

Research Review

This review is based on the Deep Blue

(<https://pdfs.semanticscholar.org/ad2c/1effcd7c3b7106e507396bdaa5fe00fa597.pdf>) paper.

Paper's Goals/Techniques

The goal of the paper is to describe the Deep Blue computer chess system, specifically the second version of the system, which defeated world chess champion Gary Kasparov in 1997. This latest version of the system was a culmination of the evolution of previous systems made to create a top-class chess-playing agent.

Deep Blue was a massively parallel computer system, specifically designed to do chess-game tree searches in a very efficient manner. This system had several important characteristics that allowed it to perform so well against Kasparov:

1. An 8000-feature evaluation function (1600 more features than its predecessor). The new additional features had been devised after observing past behaviors by Kasparov himself and Grandmaster Joel Benjamin during test games.
2. New hardware and software to radically improve search speeds to 2–2.5 million positions per second.
3. The introduction of new software tools for debugging, evaluation-function tuning, and visualization tools. All of this helped with much better match preparation.

The system was organized in three layers. One of the processing units was assigned as the master, whereas the rest as workers. The system worked by dividing the tree search into 3 different levels of depth, with the top level being handled by the master unit, which in turn delegated further inspection of leaf nodes to the 2nd-layer units. Those units processed the next set of nodes in the tree, and eventually delegated the inspection of the rest of the leaf nodes to the units in the last layer.

Deep Blue not only used classical game-search techniques like quiescence search, iterative deepening, transposition tables, and NegaScout (a more efficient version of alpha-beta pruning that leverages node ordering), but it also introduced new hardware- and software-based techniques. Some of these techniques are:

- A chip-based move generation to improve attack-moves.
- Slow- and fast-evaluation functions to manage a wide range of game scenarios from easy- to very-complex.
- Hardware-based null-window search controls with repetition detection, which eliminates the need to pass alpha/beta parameters when applying alpha-beta pruning.
- Specific software-based search algorithms to manage particularly complex chess tactics.

All of these techniques were applied using a highly parallelized decision mechanism. Finally, Deep Blue was loaded with 4000 expert-predefined opening moves, and access to a Grandmaster game database to assist with a broader spectrum of opening moves, and even with endgame moves.

Paper's Results

Many factors contributed to the success of Deep Blue. Some of the basic elements, we've covered during our Udacity class material; however, other critical aspects were also introduced both at a hardware and software level.

Even though the overall performance of the system was remarkable, further research suggested that there were still areas for improvement. Specifically, around the parallel search efficiency, the already-available external hardware search mechanism, and even further addition of pruning techniques and finer tuning of the evaluation function. Time constraints in the project forced the initial effort to leave certain system design areas unexplored. The paper encourages further research into those areas.