Research Review on Deep Blue II

Chris Kirchner

Deep Blue II success in beating reigning World Chess Champion Garry Kasparov in 1997 after its first failed attempt in 1996 relied on many hardware and software improvements from its predecessors that seemed to surprisingly depend little on algorithmic and system efficiency. While reading the article by Murray Campbell at al, "throwing the kitchen sink at it" became a predominant theme exhibited by the chess machine's designers who tended to add more functionality from previous versions, and more hardware to handle that functionality. Everything one could conceive of to use as a nascent student in Artificial Intelligence, and much more, was used to ensure victory against the top human opponent. This "everything" briefly included a multi-node massively parallel chess search engine, search extensions for non-uniform search, complex evaluation functions with tie breakers, and a game database to value searched moves. While throwing everything you got at something seems like an awful lot, the ability for a machine to beat the best human player at one of the most intelligent games before the real AI millennium was well worth it, and really, why not?

The computer architecture of Deep Blue II was as extensive as its parallel search algorithm, which utilized intimately connected hardware and software logic. The "Kitchen Sink" search engine comprised of 30 general purpose processors (IBM RS/6000 SP) ranging from 120 MHz to 135 Mhz and 480 chess chips. To put things into perspective, the average consumer computer circa 1995 was a single core 33 MHz processor with 8MB of RAM and a 1GB hard drive. Each Deep Blue processor was connected to 16 chess chips that communicated via micro channel buses and together made a node. Each chess chip was specifically embedded with chess logic to recognize up to 8000 patterns for evaluation, and even contained a database for endgame play encoded in ROM for the last 5 pieces. Worker nodes connected to a master node via a switch. The master node would start the root of the game search tree and handle the first few plys, or iterations amongst its chess chips. The worker nodes would continue processing the tree to deeper levels, distributing "leaves" of the tree to chess chips for hardware based null-window alpha-beta search, move generation, and move evaluation. Since the Master node was the computational bottleneck, load-balancing was implemented to minimize overloading, such as queuing up node tasks for each slave processor. In total, the system handled 100 to 300 million positions per second, where a three-minute search provided roughly 12 levels deep of move evaluations into a full-width game tree. Given human players can tend to go deeper with non-uniform searches, search extensions were granted for branches that seemed attractive as part of its bundle of algorithmic AI techniques.

Implemented in the hybridized software and hardware based search algorithm contained a plethora of AI techniques, many of which were embedded in the chess chips themselves. Quiescent search, iterative deepening, transposition tables to remember already visited branches, alpha-beta pruning, and something called NegaScout, were the main tactics for optimal move search. Aside from general AI search techniques, move generation and evaluation were specifically tailored for chess of course, where the move generator itself is an 8x8 array of combinatorial logic that is said to be "effectively a silicon chessboard". Since chess has a time restriction, two levels of time thresholds were introduced to keep Deep Blue at an appropriate pace, and to additionally grant extra time for moves were the previous iteration varied greatly in utility value, or when there was doubt in the quiescence. Two layers of evaluation were also implemented in the evaluation function when full evaluation became too expensive and a faster evaluation gave a reasonable approximation. Several records of best positions and moves were also utilized, with an opening book containing 4000 beginning setups and an extended Grandmaster book that was a 700,000-game database.

While the result of Deep Blue II was grand in dominating the top human opponent, the advantage seems to be mostly gained by upgrading hardware and adding more features to both hardware and software. Some work was done on evaluating or tuning the evaluation functions as well as the efficiency of the parallel search technique, but the complexity of the machine and the need to hand-tune most values made it difficult to improve significantly in this heuristics and search efficiency area. With the next current millennium, I imagine AI will be building itself with more efficiency, less of the hardware, and less of the possibly bloated features, than was needed for Deep Blue II.