Review of "Deep Blue"

The paper "Deep Blue" by Murray Campbell, A. Joseph Hoane Jr., and Feng-hsiung Hsu discusses some of the aspects of IBM Research's Deep Blue – the chess playing system which famously defeated Grandmaster Gary Kasparov in 1997 during a seminal moment in the history of the field of AI. The paper begins with a descriptive outline highlighting previous versions of chess-playing agents which had an influence on and were pre-cursors to the Deep Blue II system which defeated Kasparov – ChipTest, Deep Thought, Deep Through 2, and Deep Blue I. Design aspects which ultimately led to the success of Deep Blue 2 are introduced and emphasized in this paper. These include: use of a single-chip chess search engine, use of parallelism during game tree search, use of search extensions, a highly-complex and hand-tuned evaluation function, use of a Grandmaster game database for opening book and end-game decisions, and other components. After the successes of Deep Thought and Deep Thought 2 in tournament play, Deep Blue was designed with the intention of competitively tackling chess Grandmasters. Deep Blue II was an iterative enhancement on earlier models which ultimately led to success against Kasparov and others.

One unique aspect of the Deep Blue agent was the combined use of software and hardware game tree search within a highly-parallel system. The 480 chess chips (16 per each of 30 processor nodes connected via MPI) were each capable of searching 2-2.5 million positions per second. A single "master" node distributed search work to "worker" nodes – the master searched the top levels of the game tree, then farmed the remainder to the workers. These workers then passed off their game tree leaf positions to the specialized chess chips to complete evaluation. The chess chips featured 3 parts: a flexible finite-state-machine move generator capable of generating specialized moves, quiescence search and singular extensions with an emphasis on efficient and effective ordering, a two-pronged evaluation function capable of a fast (1 clock cycle) evaluation and a more detailed slow evaluation examining more features, and search control – making use of null-window alpha-beta search featuring a repetition detector. Finally, the chess chips featured optional extendibility with external FPGA, providing access to a transposition table, more complicated search control, and enhancements to the evaluation function. Chess chips were capable of only shallow searches, with extensions for deeper and quiescence search in certain scenarios.

New challenges presented by Deep Blue included: large searching capacity – leading to "non-uniform" search to expand certain interesting portions of the game tree but not others, use of "insurance", implementation of the evaluation function in hardware – which provided for a static evaluation time and the capability of more than 8000 features, a combined hardware/software search platform – which offers a tradeoff between flexibility on the software side and tuned execution time for the hardware end, and massively parallel search – with over 500 processors available for game tree search yet presenting a difficulty in its interaction with "selective search" and "horizon effects".

The software search featured a novel "dual credit with delayed extensions" technique that enabled fractional extensions and proper handling of "forced moves" and reactions to forced pairs of moves. This technique made use of the concept of "cashing in credit" to enable extensions in search for certain portions of the game tree while at the same time controlling "search explosion" and allowing for preservation of the "search envelop" and avoiding "oscillation". Factors contributing to assigning credit to a node included: presence of a singular better move, single legal move, or threat or mate threat, "influence" in response to an opposition's pattern of development, and other techniques. Assigning higher credit towards the root node enabled protections against search explosion. Each node is the point of control for parallelism, with parallelism possible only if certain conditions are met – using a taxonomy of type 1, type 2, and type 3 nodes with type 1 and 2 nodes serving as a synchronization point. Software search-level load balancing and "master overload" were employed to help balance the game tree search across nodes.

The context-sensitive evaluation function made use of over 8000 (sometimes multi-faceted) features, summed together to present an evaluation at the chess chip level. The paper delves deeply into the "rook on files" example of one complex feature of the game state. A tunable and well-prepared opening and extended book were present in the system to assist with start and end game scenarios – employing a database of Grandmaster and tournament play sequences and results for the system to analyze. Finally, time control was employed to maintain awareness of chess timing rules and bound search and evaluation in selecting a move and avoiding inadvertent forfeit.

The paper stresses that Deep Blue's success was made possible by a combination of its large search capability, non-uniform search, extensively hand-tuned evaluation function, and opening and extended books. However, the system could yet still be improved with the addition of FPGA, pruning mechanisms, and more effective manual and automatic evaluation function tuning in a future version.