LITERATURE RECEIVED

COMPUTER-CHESS CHAMPIONSHIP PROGRAMS: SOFTWARE DESIGN, SYNTHESIS OF EVALUATION FUNCTIONS, PARALLEL SEARCHES

by Jean-Christophe Weill[1]

Cachan, France

We have received Jean-Christophe Weill's Ph.D. thesis, accepted January 27, 1995 under the title *Programmes d'Échecs de Championnat: Architecture Logicielle, Synthèse de Functions d'Évaluation, Parallélisme de Recherche*[2]. An English paraphrase of the abstract follows.

"We summarize our work in chess programming. They cover three major items:

- a suite of programs, among which there was CUMULUS 2.0 having obtained the runner-up position to the World Champion in chess software regardless of class;
- the method of constructing evaluation functions used so as to endow our combinatorial programs with some intelligence; we also publish results of building up evaluation functions by an automated learning process;
- a new efficient scheme of parallelizing the minimax determination in a search tree.

In addition to the above, we present the state of the art of programming competitive intellectual games, the present position in search algorithms, and, likewise, the heuristics applicable to those algorithms, to parallelizing minimax, to evaluation schemes, and to the learning processes.

Programming thinking games was considered as *the Drosophila melanogaster of machine intelligence.* This area should allow techniques and algorithms to be re-applicable in other areas of artificial intelligence. Going by Shannon's words, the subject would be one in which advances would be communicated easily to the public at large. We have tackled this problem within the framework of game-playing programs having to react to a problem, which amounts to proposing a move within a limited time, which means under tournament conditions.

We present the essence of the state of the art in this programming field by exhibiting our own contribution to the subject. We compare the various minimax searches beased on α-β pruning with the NegaC* algorithm which we defined ourselves, after presenting the main results we established on the complexity of this algorithm. In the Negamax paradigm, we give a definition of the new algorithm of "of proof-number search" which we compare with our chess program ECUME when searching for mates.
We present a set of heuristics which speed up Negamax searches as well as rendering them more reliable in their choice of the right move, giving details of the options chosen in our chess programs.

After surveying all the results hitherto known in the domain of parallelizing minimax, we present the outcomes of our implementations on a distributed engine, the Connection Machine 5. These results have enabled us to define a new method of parallelization which we have compared to the best methods previously known. The comparison ranged over simulated search trees, over a competitive Othello program and over a chess program. We present advantages and drawbacks of each method.

We continue by presenting the method we chose for constructing the evaluation functions, stressing how we have achieved introducing the notion of a strategic plan into our chess programs. We also show how a program can construct an evaluation function automatically by learning. We specifically report on experiences in building a correct evaluation function for the KQKQ endgame.

Finally, we describe the full set of characteristics of our chess progams, using an analysis of their games when competing in a championship to point out their strengths and drawbacks."

---

[1] 24, Avenue du Pont Royal, 94230 Cachan, France. Email: jcw@cnam.fr.

[2] An English version has been announced by the author. Pending this, a microfiche of the thesis (187 pp.) is available.