8. AppenDix

If being selective is an act of manicuring, I plead guilty to doing so with this appendix, as I did by choosing the position of Diagram 3.

It contains the complete listing of the game between the 1993 Micro World Champion HIARCS and NIMZO-Guernica, Micro WC Munich, 1993. Each move is followed by NIMZO’s computation time in seconds. These times include operator time. Moves characterized by NIMZO as easy are in bold type. Also the moves 16. Bf2, 17. Bd3 and 18. Qxd4 hint that HIARCS uses some kind of simple-move technique. Strange is 45. fxe4: HIARCS spent about 6 minutes on the only way to recapture the piece. The programs have different styles of playing, therefore the rate of correct guesses is unusually low. Both played their first 11 moves from their opening books.

According to Hyatt (1984), it is wise to spend more time on the first moves than on those in the endgame. I think this is erroneous. NIMZO, by design, uses in the endgame about the same time per move as in the middle game.


LITERATURE RECEIVED

EIN KORREKTES PROGRAMM FÜR DAS ENDSPIEL KÖNIG UND BAUER GEGEN KÖNIG UND BAUER

Institutsbericht Nr. 36, February 1994

Wilhelm Barth

Institut für Computergraphik
Technische Universität Wien
A-1040 Wien, Karlsplatz 13/186
Austria

We paraphrase the abstract:
A correct program for the endgame KPKP has been developed using the rule method (Barth, W. and Barth, S., 1991, 1992). This method yields for every position an interval guaranteed to contain the correct evaluation. Though after a short time the program may give only a wide-range interval as a result, e.g. [drawn, won] failing to define the result of the game, it will deliver a point interval, e.g., [drawn, drawn], after sufficient time. The program has been validated by the method described in the paper cited, i.e., it has proved that its results are correct for all KPKP positions.

Experience shows that only easy rules, well-known to every skilful chess-player, had to be transferred into algorithmic form to get a program that finds the final result in a short time. One interesting and surprising rule has been inspired by Reti’s famous problem. The behaviour of the program has been tested by solving this and other problems and by analysing some positions discussed in the literature.
References


ARTICLES PUBLISHED ELSEWHERE

THE MULTI-PLAYER VERSION OF MINIMAX DISPLAYS GAME-TREE PATHOLOGY


We reproduce the abstract:
"It is widely believed that by searching deeper in the game tree, the decision maker is more likely to make a better decision. Dana Nau and others have discovered pathology theorems that show the opposite: searching deeper in the game tree causes the quality of the ultimate decision to become worse, not better. The models for these theorems assume that the search procedure is minimax and the games are two-player zero-sum. This report extends Nau's pathology theorem to multi-player game trees searched with maxn, the multi-player version of minimax. Thus two-player zero-sum game trees and multi-player trees are shown to have an important feature in common."

PROOF-NUMBER SEARCH


We reproduce the abstract:
"Proof-number search (pn-search) is designed for finding the game-theoretical value in game trees. It is based on ideas derived from conspiracy-number search and its variants, such as applied cn-search and αβ-cn search. While in cn-search the purpose is to continue searching until it is unlikely that the minimax value of the root will change, pn-search aims at proving the true value of the root. Therefore, pn-search does not consider interim minimax values.

Pn-search selects the next node to be expanded using two criteria: the potential range of subtree values and the number of nodes which must conspire to prove or disprove that range of potential values. These two criteria enable pn-search to treat efficiently game trees with a non-uniform branching factor.

It is shown that in non-uniform trees pn-search outperforms other types of search, such as α-β iterative deepening search, even when enhanced with transposition tables, move ordering for the full principal variation, etc. Pn-search has been used to establish the game-theoretical values of Connect-Four, Qubic, and Go-Moku. There pn-search was able to find a forced win for the player to move first. The experiments described here are in the domain of Awari, a game which has not yet been solved. The experiments are repeatable for other games with a non-uniform branching factor.

This article describes the underlying principles of pn-search, presents an appropriate implementation, and provides an analysis of its strengths and weaknesses."