

AWIT: A selective search chess program

T.A. Marsland
University of Alberta
EDMONTON T6G 2H1

In recent years most successful chess programs have relied on the efficiency of the alpha-beta algorithm, in conjunction with iterative deepening, to perform the equivalent of exhaustive search to whatever depth is possible within an acceptable time limit. At each new position the moves are weakly ordered on the basis of computationally cheap criterion.

In AWIT a search is initiated with a short list of candidate moves, selected on a basis of chess knowledge incorporated into the program. Part of this selection process involves a study of all checks and captures at the first two plies. As the search progresses, and a candidate refutation is established at ply N, moves at ply N-1 which explicitly deny that candidate are sought. Thus although the move lists are strongly ordered, through a detailed static analysis of the position, continuous dynamic reordering is done to find denials. The Principal Variation Search (PVS) algorithm is used, even though not theoretically valid since forward pruning is applied to limit the absolute width of search. In effect the first variation is searched with a full window, while a minimal window is used in an attempt to refute the balance of the moves.

Because the analysis of each position is expensive a hash (transposition) table is employed to store positions during the search. Management of this table is quite complex, because positions on the candidate and alternate variations are linked together with a tree structure of pointers. When the available space is too small to accommodate the expected size of the next subtree, the least valuable alternative variations are unlinked. Thus although the transposition table has two parts, the linked and the available entries, both portions contain valid positions, and positions can be successfully retrieved from either. However, new positions are only inserted onto the available space portion.

The opening book consists of about 11,000 positions, each containing one or more feasible moves. These positions are retrieved from the file system using the same hashing function that accesses the transposition table. AWIT itself is a large AlgolW program of 120 procedures, occupying about half a million bytes of primary memory for its code and data space. The program is also quite slow, examining about 8 positions per second on an Amdahl 470/V8 and looking at a tree of less than 10,000 branches per move. Statistics for the transposition table are quite variable, but an average of about 10% of the active elements in the table are salvaged from move to move as a

linked tree. Also, between 5% and 30% of the elements in the table are used in lieu of subtree searches from ply to ply, most coming from a relinking to discarded variations. The table size itself is an input parameter, but defaults to 100 entries for speed games and 600 entries for tournaments.

The name AWIT arises from the fact that in the early days of the program's development the moves selected were sometimes quite unbelievable, laughable, almost comic, in some sense witty.

Most commercial programs are very compact, use a simple design -- search every move and every response to a move and so on until some time limit is exceeded. The position reached after every sequence of moves considered (the length of this sequence is termed the effective horizon of the program) is evaluated according to some simple (computationally cheap) criteria. Because a large number of these positions must be analysed, only a modest amount of knowledge about chess itself can be included. Consequently the style of play is relatively simple, unsophisticated with few serious alternatives. Such programs tend to exchange material too readily, and are rather aimless in the endgame, since the consequences of early decisions become apparent beyond the program's horizon. All chess programs rely on the fact that most moves can be refuted for obvious reasons, and this notion is incorporated effectively into the search method, the alpha-beta algorithm. Even so, the number of positions examined by the method is very large, so either very fast computers must be used or the style of play must be relatively simple.

With the selective search approach used in Awit, emphasis is placed on preliminary searches to discard moves that are thought to be irrelevant. However, some preliminary searches are done to identify simple sacrifices and forks (multiple attacks). An important point is that the moves discarded are not permanently lost, but can be reconsidered if they have properties that may refute an opponent's "killer" move (e.g., an apparent mate). This analysis is quite complex and time-consuming, but even so makes it possible, in principal, for the program to search deeper along the selected line. On the other hand discarding moves (even temporarily) has been likened to playing with half a mind, or with one eye closed, since good moves may exist amongst those ignored. Also, the program itself is much larger, more complex and contains far more chess knowledge than a brute force program. In particular, it must be concerned with the placement of pieces and their inter-relationships, in order to ensure that a defensive structure exists that cannot be violated for several moves. By this means an attacking variation can be developed without the need to constantly examine defensive measures. Selective search programs tend to play more complex games, do not exchange material so readily, and have more knowledge from which to develop long-term piece movement in endgames. Even so, they are prone to gross oversights.