Recent Developments in Computer Chess

by

D. Kopec[†]
Machine Intelligence Research Unit
University of Edinburgh
Scotland, U.K.

"If one could devise a successful chess machine one would seem to have penetrated to the core of human intellectual endeavour."

A. Newell, J. C. Shaw and H. A. Simon (1963)
"Chess playing programs" in Computers and Thought

The above was written fourteen years ago and since that time it must be admitted that progress has been disappointing.* However some recent developments are encouraging:

(1) a. Northwestern's Chess 4.5 won the Class B section (USCF rating scale 1600-1800) of the Paul Masson Tournament, held in July in California (5 - 0).

b. Chess 4.5 also defeated a Class A player in an individual match game (see page 41 for game record).

(2) Special chess hardware with a 10-ply lookahead has been designed by Greenblatt and Missouris. Two Experts have been included among its victims. It can look at more than 100,000 boards/sec.

Even in the end-game, generally agreed to be the most difficult phase of chess, there have been some successes. In the Soviet Union, the entire space of all R + P vs. R positions has been computed out and stored as a lookup tabl[e] Many masters and grandmasters have been known to go wrong in such endings. Th[e] same was done with Q + P vs. Q. David Bronstein, a former World Championship contender, consulted that database for the correct strategy to win an adjourne[d] game.

The following notes concern interesting features of two chess programs which I have myself been concerned with. Not all the ideas were implemented.

Work at Dartmouth College, New Hampshire, USA

The computer chess project at Dartmouth is headed by Dr. L. Harris. Work-ers have included W. Montgomery, H. Terrie, D. Levner and myself, among others The program was written in GCOS, the assembly language for the Honeywell 635. Dartmouth's program was the first program to challenge Northwestern's

---

[†]U.S. National Master

*This article was written before CHESS 4.5's later successes.

ascendancy in the ACM tournaments. In the 1973 ACM U.S. Computer Chess Championship, held in Atlanta, Northwestern's program was lucky to draw against Dartmouth. It only did so because the latter had no repetition check. Particularly encouraging was the fact that in a game of more than 50 moves, Dartmouth was never in a losing position.

The program is divided into two major evaluation functions, $\hat{h}$ and $\hat{d}$. $\hat{h}$ is concerned with the "soft", positional features of a given board position, while d is concerned with the "hard" tactical features of a position. The specific chess concepts which comprise $\hat{d}$ and $\hat{h}$ are called "Detectors". A set of related detectors are assigned various values (weights) and are put into a table. $\hat{h}$ includes tables such as Centre Control, Piece Mobility, Pawn Structure, King Safety, etc., while $\hat{d}$ includes tables such as Pins, Forks, Discovered Attacks, Levers. The program is also divided into modules (Opening Middle Game, and Endings) which allow greater flexibility in the assignment of weights. For example, in the opening, Piece-devélopment, Centre control, and King safety are stressed. A persistent problem which many programs still have is the too early development of the queen, because of its tremendous square control, mobility, and ability to produce threats. By assigning a value of -300 (where 100 = pawn) to every minor piece (B or N) still on the back rank, piece development is given prominence, since the program tries to get rid of these initial negative values. Other examples of tables which employ modular flexibility are Occupation of the Centre, and Rook on 7th. Greater weights are assigned to these in the middle game and ending than in the opening, to avoid moving the same pieces too often, before others have moved at all.

An idea which was never fully implemented was that of an "Attack-Defence Ratio". This is a measure of the difference between the sum of the forces attacking the quarter of the board where the enemy king is located and the sum of those forces which defend the same squares. If this difference in force is greater than a certain threshold value, an "alarm" is set off which results in a higher $\hat{d}$ value and an increase in the depth of search. In this manner, long sacrificial variations are more carefully investigated. A benchmark of sacrificial positions would be a good test for its effectiveness.

Dartmouth's most "informed" table was the one on pawn formations, called "PFORM". Among its standard detectors were Isolated Pawns, Backward Pawns, Doubled Pawns, Passed Pawns, and Duos. Detectors such as Chains, Mini-chains, Shielded Backward Pawns, Potential Passed Pawns, and the table, "Levers", were among the more esoteric concepts which were added later. Many of these definitions were taken directly from Hans Kmoch's classic work <u>Pawn Power</u> (<u>see</u> Kmoch 1959). The concept, Levers, using a modified definition of my own--"pawn moves which improve our formation and hurt our opponent's"--proved useful in the recognition of critical pawn moves. In addition, the levers concept helps to guide the placement of pieces especially in the opening and middle game. It could also help toward plan formation. Some further pawn formational concepts from <u>Pawn Power</u> which were never programmed were Outposts and Weak Square Complexes. The Dartmouth program is probably, in theory, capable of more sophisticated pawn formational evaluations than any other program; however the implementation is rudimentary. The program had at one time approximately 50 detectors in various tables and many others were planned.

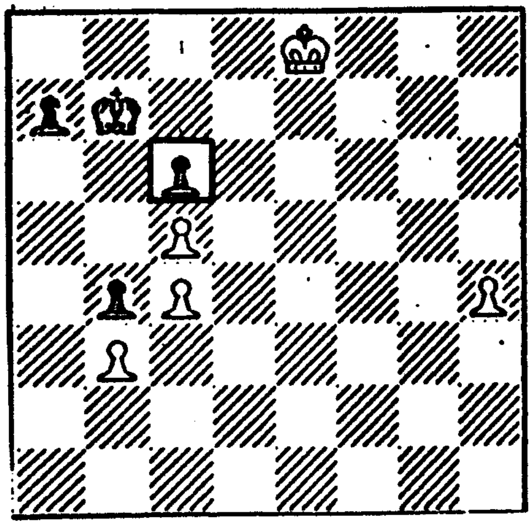


Figure 1: From Basic Chess Endings (R. Fine)

Thus, 33[1,31], i.e. pawn on 33 is a Counterpawn to pawn on 31
33[2,32], i.e. pawn on 33 is a Ram to pawn on 32
33[3,21], i.e. pawn on 33 is a Sentry to pawn on 21
33[8,22], i.e. pawn on 33 is a Potential Protector to pawn on 22.

(See preceding Table for definitions.)

Rather than queening the passed RP immediately, which would result in a quick draw due to Stalemate, the solution lies in White's capture of that pawn on 33.

A graph representation for the same position is given below, as Figure 2, using Tan's notation as in Table 1, with the addition of "Λ" to denote a passei (my notation).

Another concept put forward by Tan is that of an ADD (Attack Defense Diagram). (See Figure 3.) Some of the evaluations suggested for an ADD are as follows: (1) Relations within fronts; (2) Defenses to threats of (1); (3) Possible attacks of Kings agains Pawns; (4) Defenses to (3); (5) Support possibilities; (6) Joint attacks. Tan also breaks down nearly all possible relationship that might go into the ADD into B.N.F. (Backus-Naur Form). As yet there is no working program for the ADD. As a step towards designing one I have tested how the evaluations would work on some simple K + P endings. It seems that at leas three concepts must be added: (1) Opposition, (2) Triangulation, (3) Outside Passed Pawns. These features are important, and occur frequently.

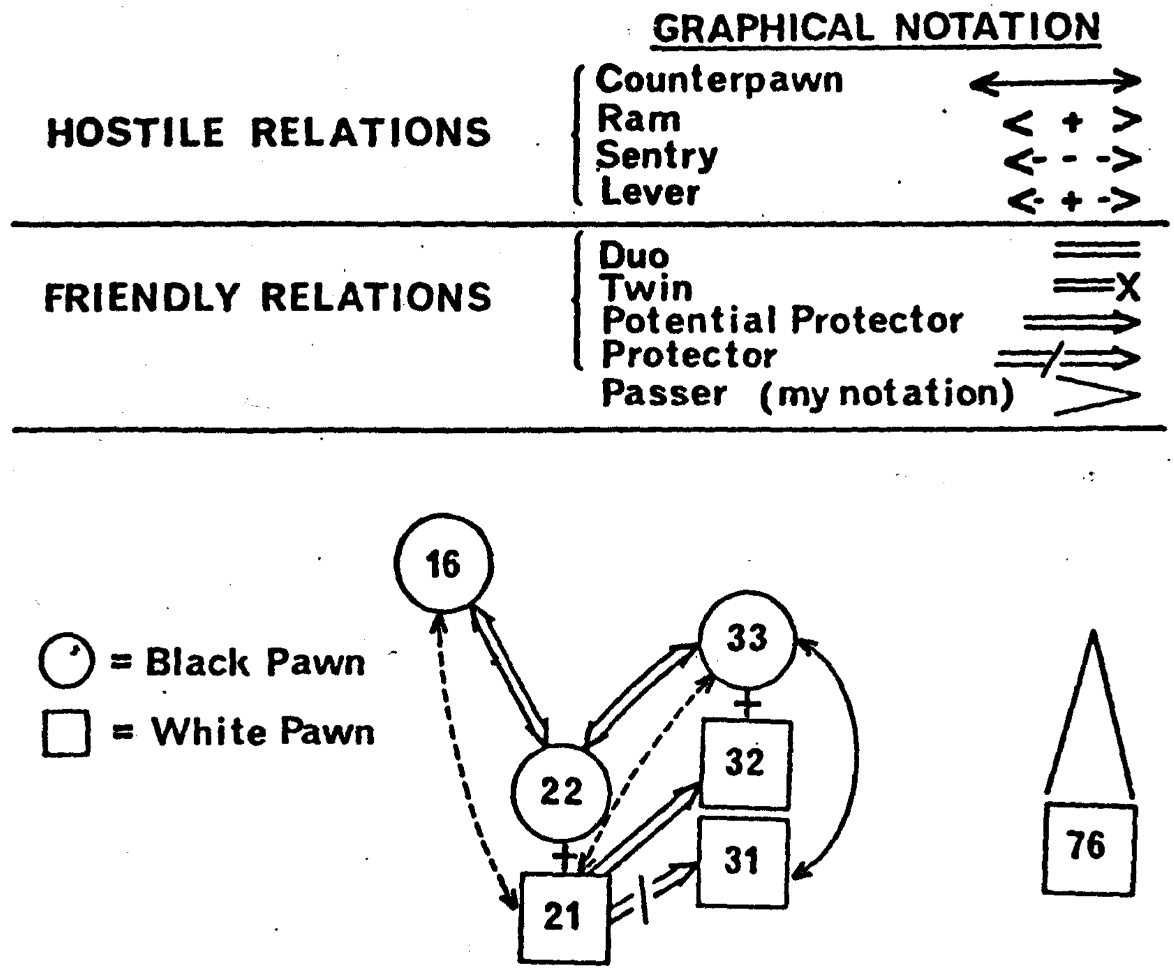
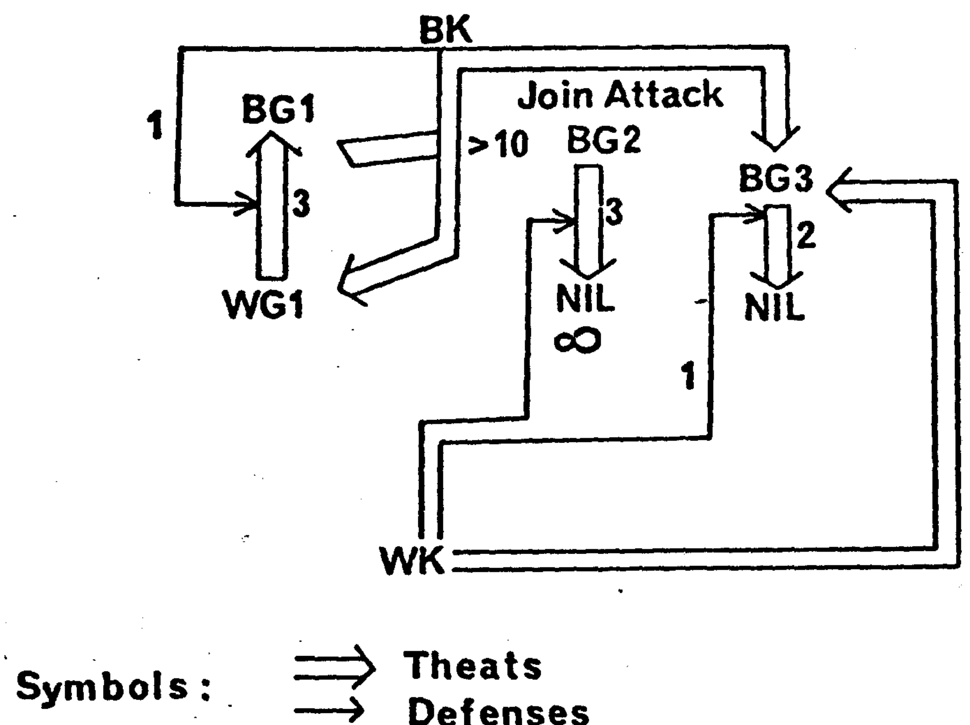


Figure 2: Graphical representation for chess position shown in Fig. 1

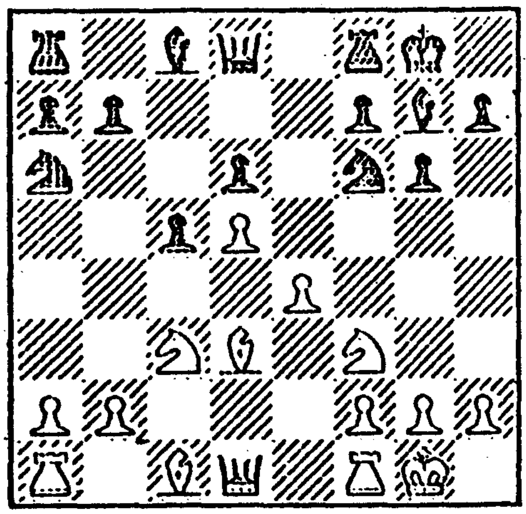
TAN'S ATTACK DEFENCE DIAGRAM



Common Faults of Existing Programs

One obvious fault is due to what Berliner has termed "the Horizon effect". That is to say, given a fixed depth of lookahead, and the usual definition of quiescence (no checks or captures) a program may waste time and material in order to defer a loss which to a deeper search can be seen as inevitable; or, alternatively it grabs shortsightedly at a small gain, oblivious to a major one beyond its horizon. It might be helpful to consider all material gain threats in the same league as captures and checks. In other words, a position is quiescent when there are no checks, captures, or threats to gain material. Another way to deal with this problem might be with a one-ply search beyond all double threats. However, a precise definition of a double threat is not easy for the programmer.

Other program faults are due to its lack of knowledge. We cannot expect too much from computer chess play when there is so little to guide its decision making process. The human chess player, particularly at the master level, employs rapid knowledge-based pattern recognition or feature detection. There are also many experiences (good or bad) which cause him to revise his thinking. Such experiences lead to the formulation of a hierarchy of rules which can guide play. Consider the following position which might easily occur in the Benoni Defense.



Black has just played N-QR3.

Most computers would probably play B X N here, if out of their opening library. On pawn structural grounds alone the move is very reasonable (Black's isolated doubled Queen's Rook Pawn). However experience has shown that Black's resulting two Bishops and half-open Queen's Knight file more than compensate for the weak-looking pawns. Another piece of knowledge which most programs lack is when the qualitative value of pieces changes. When is a Bishop worth more than a Rook? When is a Knight worth more than a Bishop? Many such conceptualizations over a chessboard are based on the style and originality of individual players.

Representation of Human Chess Knowledge

I have only recently been convinced by Professor Donald Michie's argument that every "bit" of chess information assimilated by Bobby Fischer over 34 year of life can in principle be input into a computer. Thus, if we consider that the fastest rate of human information input is 30-50 bits/sec. and multiply thi by the seconds in a year and then by 34 (Fischer's age) we still have a quantit of information which can easily be stored and retrieved by a computer. Therefore the problem is one of representation, not of space.

When we consider that at most a chess master will "look at" (search) 50 boards versus the more than 100,000/sec. which can be viewed by Greenblatt's new chess machine, Cheops, we see the need for some rules of compression of human chess knowledge. Such rules as: two bishops are an advantage, knights are better than bishops in closed positions, don't block the Queen's Bishop Paw with the Queen's Knight in Queen's Pawn openings, and the continuous refinement of such rules by human players must surely strengthen any computer chess progra But how many of the rules by which masters play we should aim to use in a progr is another question. The number has been estimated by two different sources, Simon and Gilmartin (1973) and Nievergelt (1977) as lying in the range of 10,00 to 100,000.

Challenge to Reader

As a tail piece, I wish to put forward an hypothesis that a chess game between two computers can easily be distinguished from one between Bobby Fische and a master. It is based on the general statements about computer play made i this paper and also the following points:

(1) Integration of Position (Entropy)
(2) Development of Pieces
(3) Material Equality
(4) Simplification.

I suggest the reader take a few minutes to look at each of the positions i Figures 4-9, and then decide for each whether it is between two computers, or Fischer vs. a Master. Each of the positions occurred after 20 moves of play in the game. The solutions are on page 46.

Bibliography

Berliner, H. J. (1974) Chess as Problem Solving: The Development of a Tactics Analyzer. Ph.D. Dissertation, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA.

Fine, R. (1941) Basic Chess Endings. Philadelphia: David McKay; London Bell.

Harris, R. L. The Heuristic Search and the Game of Chess - A Study of Quiescence, Sacrifices and Plan Oriented Play. In Proceedings of IJCAI 1975, Vol. 1, Tbilisi, Georgia, USSR. pp. 334-339.

Kmoch, H. (1959) Pawn Power. New York: David McKay Company, Inc.

Levy, D.N.L. (1975) The 1975 U.S. Computer Chess Championship. Computer Science Press, Inc., Woodland Hills.

Newell, A., Shaw, J. C. and Simon, H. A. (1963) Chess playing programs. In Computers and Thought (eds. E. A. Feigenbaum and J. Feldman) pp. 39-70. New York: McGraw-Hill.

Nicvergelt, J. The information content of a chess position, and its implication for the chess specific knowledge of chess players. SIGART Newsletter (in press).

Simon, H. A. and Gilmartin, K. (1973) Cognitive Psychology 5. 29-46.

Tan, S. T. (1977) Describing Pawn Structures. In Advances in Computer Chess 1 (ed. M. Clarke) pp. 74-88. Edinburgh: Edinburgh University Press.

Wade, R. G. and O'Connell, K. J. (1972) Bobby Fischer's Chess Games. Garden City, New York: Doubleday & Company, Inc.

-ooOoo-

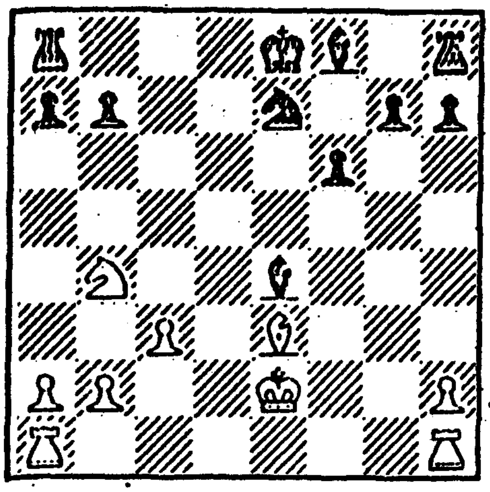


Figure 5

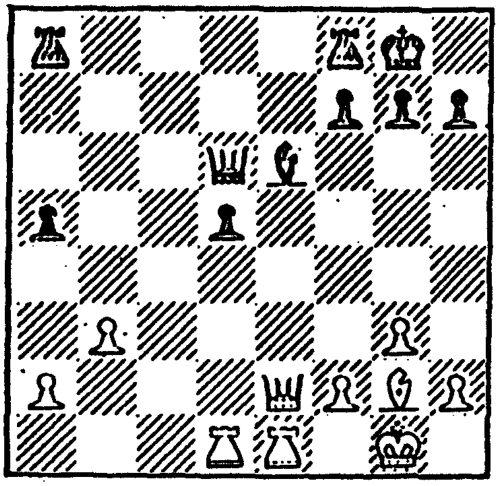


Figure 6

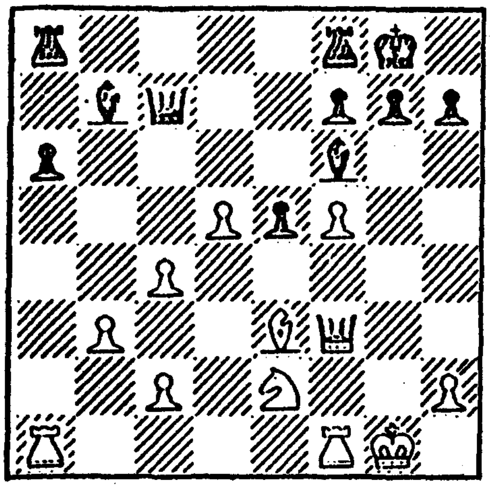


Figure 7

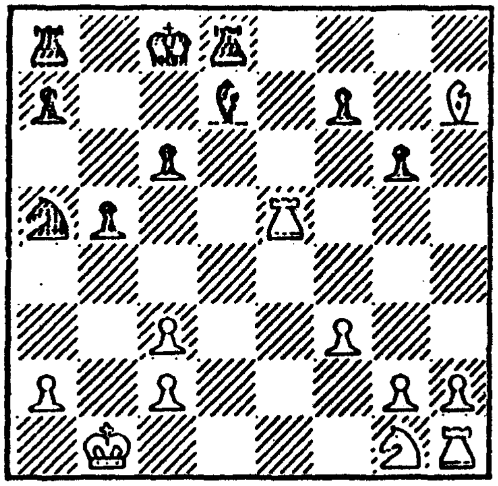


Figure 8