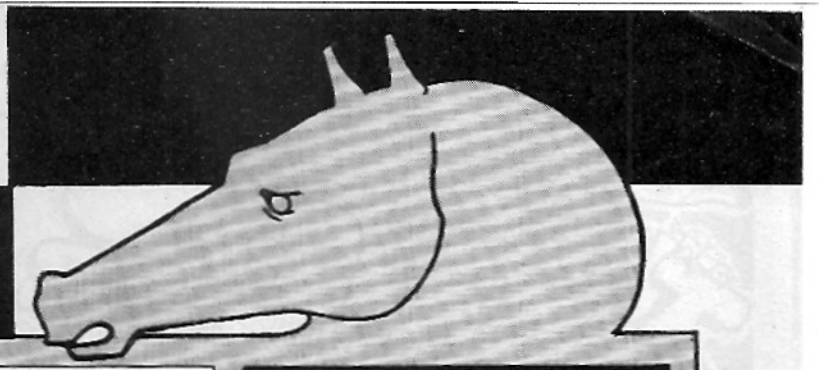
CHESS



Computers are well known to be weak at positional play, the intricate manoeuvring for a small advantage, such as a well-placed Knight. However, they also have problems with tactical play, even with what seems like the simple task of finding forced checkmates in positions where they are known to exist.

Look at figure one. This is an example of a "White to play and mate in two moves" problem of the kind often found in newspapers and magazines.

The first point to note here is that the specification for White to mate in two moves has no practical value. Chess problems are not game positions but artistic compositions on the 64 squares of the chessboard.

Solving a chess problem is not a matter of examining every possible sequence, but of identifying the composers' *theme* in setting the problem.

The use of a computerised "mate in two" program using exhaustive analysis sadly undercuts the aesthetic experience of problem solution.

Simply by considering every possible legal combination of two moves for White and one for Black, the program inevitably reveals the winning move. (The reader is left to discover the solution for figure 1 using either a computer or a human brain, as preferred.)

In finding checkmating sequences in positions that arise in practical play, very different methods need to be applied.

The longest variation in a checkmating sequence arising from play can easily be 7 or 8 moves (of each side) deep, so analysing every possible combination as a general solving method if completely out of the question.

On the positive side, the moves in a checkmating sequence from play are usually fairly obvious ones of a forcing nature, such as checks and captures (or sacrifices of material) whereas those in composed problems are frequently "quiet" or unlikely moves, such as a Bishop move one square further away from the scene of battle, which are difficult to perceive even for strong over-the-board players.

The MATER program of George Baylor and Herbert Simon which appeared in the mid-sixties was an early pioneering attempt aimed at finding forced checkmates in game positions, given that they are known to exist, of necessity employing "Artificial Intelligence" rather than "brute-force" means.

Figure 2 is a fairly straight-forward example of the positions solved by the first version of the program, MATER 1.

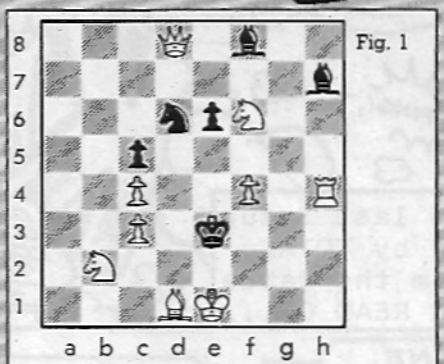


Fig. 1

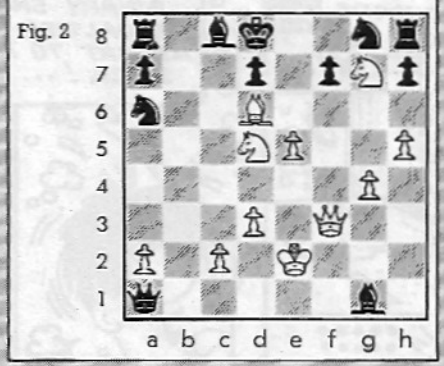


Fig. 2

The major difficulties with any such program are deciding what moves to consider and in what order, and at any point in the analysis deciding what to examine next.

Deciding when to stop analysing a line is probably the hardest task of all. A badly written program might spend many hours, days, even longer, looking at variations beginning 1. P-QR3, Q-R7; 2. K-Q2, Q-R8; 3. P-R4, etc., which humans would discount at a glance.

MATER's choice of moves to examine is simple but works quite well. At any move for White (assumed to be the winning side), only checks are considered and at every White move preference is given to the check which leaves Black the fewest legal moves in reply.

As will be seen below, this may involve abandoning the current line of analysis and starting another.

Any White moves which give Black more than four legal replies are automatically discarded. Any remaining choice amonst White moves is resolved by giving priority to double checks, then to checks with no capturing replies.

When it is Black's turn to move, all legal moves must be considered, but there is a considerable advantage in looking first at those moves which are the most likely to refute White's attack.

For this reason, MATER looks first at Black moves which capture as valuable a White piece as possible. If there is still a choice, King moves get priority.

This is how MATER solves the position in figure 2.

1. Generate all White checking moves. 1. Q-B6ch has two legal replies. 1. N-K6ch has 3 and 1. B-B7ch and 1. B-K7ch each have one.

2. Choose. 1. B-K7ch (arbitrarily) from the last two for analysis, on the basis of minimising Black's replies.

3. Generate Black's forced reply 1. . . .NxB.

4. Generate all White's checking moves. 2. N-K6ch is the only one and Black has three legal replies.

5. Choose a White move to examine next from those not yet considered. Candidates are 1. Q-B6ch (two replies), 1. N-K6ch (three replies), 1. B-B7ch (one reply) and 2. N-K6ch (three replies).

The 'best' one is 1. B-B7ch so the previously played moves (1. B-K7ch, NxB) are retracted and 1. B-B7ch is played instead from the original position.

6. Black's move is again forced, 1. . . . NxB.

7. Generate White's checking moves. 2. N-K6ch (four legal replies) and 2. Q-B6ch (two replies).

8. Choose a move to consider next, from 1. Q-B6ch (two replies), 1. N-K6ch (three replies), 2. N-K6ch (four replies) and 2. Q-B6ch (two replies).

Since 1. Q-B6ch was generated first, it is chosen in preference to 2. Q-B6ch (both with two legal replies). So 1. Q-B6ch is now played in the *original* position.

9. Black now has a choice of two replies 1. . . . NxQ and 1. N-K2. The capture 1. . . . NxQ is tried first.

10. Generate White's legal moves; these include 2. B-K7 mate, which has no legal replies at all and is accordingly chosen to look at next.

11. Since 1. . . . NxQ was unsuccessful, Black's only other legal reply to 1. Q-B6ch is tried, namely 1. . . . N-K2.

12. Generates White's legal moves; these include 2. BxNch which has no legal replies and is chosen to consider next. Since it has now been found that neither 1. . . . NxQ or 1. . . . N-K2 avoids checkmate, White's winning first move is established to be 1. Q-B6ch.

Although figure 2 is quite a simple position and one that, in isolation, could have been solved by an exhaustive "mate in two" program, the same method embellished to consider other forcing moves as well as checks for White can solve many other problems for which an exhaustive search method would be completely inadequate in any reasonable amount of time.

BY MAX BRAMER