To lesser souls who have difficulty remembering their own telephone numbers, the grandmasters of chess seem intellectual prodigies, who perform feats of memory and discovery unachievable by ordinary mortals. The great chess players are also a puzzle to psychologists, who find it difficult to reconcile these exploits with current theories about the problem-solving process. This paper attempts to clear away some of the mythology which surrounds the game of chess by showing that successful problem solving is based on a highly selective, heuristic "program" rather than on prodigies of memory and insight.

TRIAL AND ERROR SEARCH IN SOLVING DIFFICULT PROBLEMS: EVIDENCE FROM THE GAME OF CHESS

by Herbert A. Simon and Peter A. Simon

Carnegie Institute of Technology and Grove School[1]

THE game of chess provides often-cited examples of insightful discovery and prodigious human memory. Grandmasters frequently "see" decisive, winning moves whose force is not obvious to weaker players even after the moves have been pointed out to them. A number of chess masters can play many games simultaneously without sight of the board. Since the possible lines of play on a chessboard increase geometrically to astronomical numbers (e.g., one million possibilities, on the average, if the position is analyzed only two moves deep for each player; one billion possibilities, three moves deep), these feats of memory and discovery pose a problem for theories that would seek to explain human thinking and problem solving in terms of relatively simple processes operating in real time.[2]

Mating combinations—series of checking moves that end in a mate the opponent cannot escape—provide much of the spectacular in chess, the "brilliancies" comparable to the final smashing charge of an army that has first moved into position. In this paper we shall examine a number of mating combinations, including two of great historical renown, in order to measure the amount of search and memory capacity required to discover the combinations in over-the-board play. In this way, we shall arrive at some quantitative estimates of the processing and storage requirements for problem-solving achievements that are viewed as lying at the limits of human ability—acts regarded by connoisseurs of chess as highly creative.

We have observed that looking ahead even three moves in a chess position, considering each legal possibility for the players in turn, on the average calls for the exploration of a billion branches of the game tree. Nevertheless, some of the recorded mating combinations are as many as eight or more moves deep (that is, eight moves for each player), and it is these that give rise to much of the mythology that is current about the mnemonic and visualizing powers of grandmasters. Given the known facts about the proliferation of the analysis tree—perhaps 10^{24} branches in eight moves—how can a man see ahead so far and remember so much? We shall try to show that a man cannot and need not in order to discover such combinations.

Our central hypothesis is that the behavior of a chess player in searching for a mating combination is governed by a *program* that

[1] This investigation was carried out as part of the Carnegie Tech-RAND research on cognitive processes and complex information processing. We have had numerous helpful discussions with our colleagues G. Baylor, A. Newell, and J. C. Shaw. We are grateful also to the Carnegie Corporation for a grant in support of this work.

[2] An excellent analysis of chess playing from a psychological standpoint will be found in de Groot (1946). The theoretical position of the present paper stems from Newell, Shaw, and Simon (1958) and the literature cited there.

determines which moves he will consider and which branches of the game tree he will examine. We use the term "program" exactly as it is used in the digital computer field, to denote an organized sequence of instructions, executed serially in a well-defined manner. We shall describe in detail a specific program for discovering and verifying mating combinations that is powerful enough to discover a great many such combinations, including some of the most spectacular in chess history. We will show that a player "programmed" in this way would discover these combinations with a moderate amount of search and relatively modest demands on his memory capacity. The program is, in fact, so simple in construction and execution that we have been able to test it by hand simulation, carrying out its instructions step by step without recourse to a digital computer. We do not argue that the program we shall describe is exactly like that followed by any chess player, but that any one of a number of programs of equivalent power could account for the combinatorial prowess of chess grandmasters. Hence the program provides at least rough quantitative estimates of the magnitude of these problem-solving tasks.[3]

THE MATING COMBINATIONS PROGRAM

The basic idea of the mating combinations program is that the tree of possible moves must be examined in a highly selective fashion, and not exhaustively. Three principles govern the selection:

1. The attacker only examines moves that are "forceful"—the specific program we shall describe only examines checking moves. Since the attacker is seeking a line of play that leads to checkmate, he is under no obligation to examine all the moves legally available to him, but only those he thinks promising.

2. All legal alternatives open to the opponent, when it is the opponent's turn to move, must be explored. The essence of a mating combination is that the opponent is unable to escape checkmate no matter what he does.

3. If any move the attacker examines, no matter how forceful, allows the opponent numerous moves in reply, the attacking move is abandoned as unpromising. This principle has a double function. First, it reduces the size of the tree of alternatives that has to be explored. Second, restricting the freedom of action of the opponent is an essential aspect of entangling him in a mating net he cannot avoid. Hence, the fact that a move allows few replies increases both the likelihood that it will lead to a mate and the feasibility of tracing out its consequences.

The mating combinations program can now be described very briefly. The program generates all checking moves for the attacker, and lists them in priority order on the basis of the following criteria:

A. Give highest priority to double checks (moves that attack the opponent's King with two or more pieces simultaneously) and discovered checks (moves that take another man out of a piece's line of attack on the opponent's King).

B. Check with a more powerful in preference to a less powerful piece.

C. Give priority to checks that leave the opponent with the fewest replies (don't consider interposition of an undefended piece a reply).

D. Give priority to a check that adds a new attacker to the list of active pieces.

E. Give priority to the check that takes the opponent's King farthest from its base.

Experiments (some of which we shall report) show that the exact priority order does not much affect average performance of the program. We consider the above criteria in lexicographical order. If two or more alternatives are tied as "best" on a criterion, we move down to the next criterion. The highest-priority attacker's move is to be evaluated as + (leads to mate), 0 (leads to material gain, but not mate), or — (does not lead to gain or mate) before the next move is considered. If the evaluation is +, analysis terminates and the move is played; if 0, the move is retained as "possible" but search continues;

[3] The methodological issues on which this approach to the study of human problem-solving is based are discussed in Newell and Simon (1961).

if —, the evaluation is kept but the move rejected.

In the evaluation, all the opponent's legal replies are considered, in a priority order based on gains in material and increase in King's mobility. The highest priority defense is explored until it can be evaluated, then the next, and so on.

At any point at which the attacker has no further checks, or his opponent has four or more legal replies, analysis terminates with a value of 0 or —. At any point at which the attacker checkmates, the analysis terminates with a value of +.

A number of strong chess players—including one former grandmaster, a master, and an expert—who have examined this program agree that it incorporates an important part of the heuristics they use in discovering mating combinations. Certain heuristics well known to chess players are missing from the program, however. In particular, good chess players do not limit their search entirely to checking moves, but examine also certain other forcing moves—for example, attacks that threaten mate in one move and sacrificial moves that weaken the pawn protection of the opponent's King. Hence, the program undoubtedly underestimates selectivity of a chess master's analysis program, and probably exaggerates the amount of search required to discover and evaluate strong moves.

HAND SIMULATIONS

The 136 positions discussed in the chapter on mating attacks in Fine (1952) provide material for studying the performance of the program by hand simulation. Taking one of these positions, we use the program rules to search for the mate, recording each of the alternatives that is examined, and thus build up a "tree" of possibilities. The positions in Fine were not used in constructing the program.

It appears that the program will discover the combinations in about 52 of the 136 situations—for the mates in these situations all come from sequences of checking moves. Ten more combinations would be discovered if the 1-move mating threat were added to the list of possibilities explored by the attacker.

Table 1 provides an evaluation of the exploration required in four cases, including two that were undoubtedly among the most difficult. To interpret the data, we have to distinguish two trees of move possibilities: the *exploration* tree, and the *verification* tree.

The exploration tree

The attacker has to *discover* a branching sequence of moves, one subtree of which leads to a checkmate. The discovery usually involves exploring some branches that turn out to be false leads. The exploration tree is precisely analogous to the paths tried by a subject in a maze-running experiment, except that it includes branches for defender's choices as well as branches for the attacker's tries.

The verification tree

Analyses of mating combinations, as printed in chess books, do not include the whole exploration tree, but only that part of it which is necessary to verify that the combination is valid, or sound. That is, the analysis shows a single "correct" choice at each node for the attacker, but every legal reply at each node for the defender. We shall call this tree the verification tree. It is precisely analogous to the correct path in the maze. It is a tree instead of a single path because all alternatives allowed to the defender must be tested.

Table 1 shows the maximum depth of exploration, the number of positions in the exploration tree produced by the program, the number of positions in the verification tree,

TABLE 1
EVALUATION OF EXPLORATION REQUIRED FOR FOUR MATING ATTACKS

Combination	Depth (D)	Exploration (E)	Verification (V)	E/V	V/D	E/D
37	4	69	14	4.9	3.5	17.3
39	2	9	5	1.8	2.5	4.5
141	3	77	5	15.4	1.7	25.7
152	8	62	19	3.3	2.4	7.8

and certain ratios of these quantities for the four positions mentioned above. The first of these positions is the 4-move mate in Anderssen-Dufresne, one of the celebrated brilliancies of chess history; the fourth is a well-known position from Lasker-Thomas, in which Lasker announced and delivered mate in eight moves. The inclusion of these examples demonstrates that a program no more complicated than the one described here would discover brilliant mating combinations of grandmaster stature.

If we regard depth, positions in the exploration tree, and positions in the verification tree as measures of the difficulty of these combinations, then we see that these positions are not ordered in the same way with respect to the different measures of difficulty. Between depth and the size of the verification tree there is a fairly close correlation—there are, on the average, between two and three positions per move in depth. Notice that the size of the tree varies linearly, not exponentially, with depth. This characteristic, which results from the forcing character of most of the attacker's moves, is what makes deep analysis possible in combinations.

There is little correlation between the respective sizes of the verification and exploration trees. The sizes of the exploration tree and the verification tree correspond, at least approximately, to amounts of discovery and fixation, respectively, involved in the problem-solving task (Simon, 1957).

What do the numbers in Table 1 tell us about human problem-solving processes and memory? Let us, to use a round number, take 100 as the upper limit of the exploration tree for a combination that a chess grandmaster would actually discover in over-the-board play. Since a player may ponder fifteen minutes or more in a complex situation, he would take perhaps ten seconds per position examined in the exploration tree. Since all the positions in the tree are closely related—each differs from the adjacent ones by the move of only one piece—the information contained in them is highly redundant, and the rate at which information has to be handled under these assumptions is not great.[4]

Moreover, not all the positions in the exploration tree need to be fixated by the player. Some can be stored momentarily in immediate memory and, once they are evaluated, only the evaluation and not the position needs to be retained in memory. The number of positions in the combination that need to be in memory at any one time will depend on the shape of the verification tree, and it is likely that a measure of depth could be worked out comparable to Yngve's measure of the depth of English sentences. Lacking a detailed processing model of the kind he has constructed for syntax, we can reasonably assume that the skilled chess player will so organize his analysis as to keep the depth—in terms of immediate memory requirements—within tolerable limits. Since it would not be particularly difficult for a skilled writer to produce and memorize in the course of fifteen minutes or an even shorter time an entirely grammatical English sentence of 75 or 100 words, especially if the sentence contained one or more sequences of clauses and phrases of parallel construction and similar wording, the immediate memory requirements for the chess combinations do not appear to be greater than the requirements for producing complex, grammatical prose. (The preceding sentence contains just 77 words.)

VARIANTS OF THE PROGRAM

Table 2 reports some further experiments with the mating combinations program. The program described above (which we call MCP-1) is compared with a later version (MCP-4) that has different rules to determine the order in which alternatives are to be explored. The positions in Table 2, which were taken from the problems page of the January and March 1957 issues of the *Chess Review*, are representative of the kinds of mate-in-two and mate-in-three problems that can be found

[4] Our knowledge of human information processing does not yet give us any good norms. In human speech, transmission rates of 10 to 20 bits per second are common (Luce, 1960, pp. 69-79). We know that a substantial amount of processing can occur in 10 seconds. We know also that a nonsense syllable of low association value can be fixated in about 30 seconds. From these considerations, a processing time of 10 seconds per position considered appears to be of a not unreasonable order of magnitude.

TABLE 2
COMPARISON OF MATING PROGRAMS 1 AND 4

Problem		Verification Tree	Exploration Trees MCP-1	MCP-4
Jan. 1957	1	3	10	5
	2	5	19	19
	3	3	4	4
	6	9	128	26
	9	9	24	28
March 1957	1	3	60	500+
	2	11	77	26
	4	8	12	12

in actual games (as distinguished from so-called "composed" problems that are deliberately created). Reasonably strong chess players can usually discover the solutions in a few minutes, and this is reflected in the fact that the exploration trees are generally smaller than those for the Anderssen-Dufresne and Lasker-Thomas positions.

Table 2 illustrates the effects of relatively small changes in program structure using basically the same heuristics. The revised program is superior in three cases, inferior in two. In three cases, both programs generate the same exploration trees. Each of the programs would in one case have failed to discover the combination in a reasonable computing time (taking 100 positions as the limit), and the failure occurs in different positions in the two cases. This experiment suggests that we cannot create a program uniformly better than both of these simply by permuting the order in which moves are considered.

A discussion of other experiments of the same kind we have made would be of interest from the chess standpoint, but would add nothing essential to our picture of the human problem-solving process. Our findings are consistent with the other knowledge that is available on human thinking in chess. De Groot (1946) has obtained thinking-aloud protocols from grandmasters and other strong chess players, and has estimated the sizes of the exploration trees. In a complex middle game position (not a mating position), for example, he found that five grandmasters examined 20, 21, 22, 36, and 76 positions, respectively; five experts examined 16, 17, 29, 31, and 61. The average for the grandmasters was 35, for the experts, 31. Thus, there was no significant difference in amount of verbalized exploration between grandmasters and experts. Four out of five of the grandmasters, however, and *none* of the experts found the best move in the position. Clearly the grandmasters had a more effective selective heuristic to guide their exploration than did the experts.

CONCLUSION

The conclusion we reach from our investigations is that the discovery of "deep" mating combinations by expert chess players requires neither prodigious memory, ultra-rapid processing capacities, nor flashes of insight. Combinations as difficult as any that have been recorded in chess history will be discovered by the selective heuristics we have outlined, with amounts of search and with processing speeds that do not appear extravagant in relation to the measures we have of simpler kinds of human information-processing performance. The evidence suggests strongly that expert chess players discover combinations because their programs incorporate powerful selective heuristics and not because they think faster or memorize better than other people.

REFERENCES

de Groot, A. D. *Het Denken van den Schaker.* Amsterdam: Noord Hollandsche Vitg. Mij., 1946.

Fine, R. *Middle game in chess.* New York: David McKay, 1952.

Luce, R. D. (Ed.), *Developments in mathematical psychology.* Glencoe, Ill.: Free Press, 1960.

Newell, A., Shaw, J. C., & Simon, H. A. Chess-playing programs and the problem of complexity. *IBM J. Res. Devel.,* 1958, 2, 320-335.

Newell, A., & Simon, H. A. Computer simulation of human thinking. *Science,* 1961, 134, 2011-2017.

Simon, H. A. Amounts of fixation and discovery in maze learning behavior. *Psychometrika,* 1957, 22, 261-268.

(Manuscript received June 18, 1962)