Machile Files
Box 26

CHESS REVIEW

the picture chess magazine

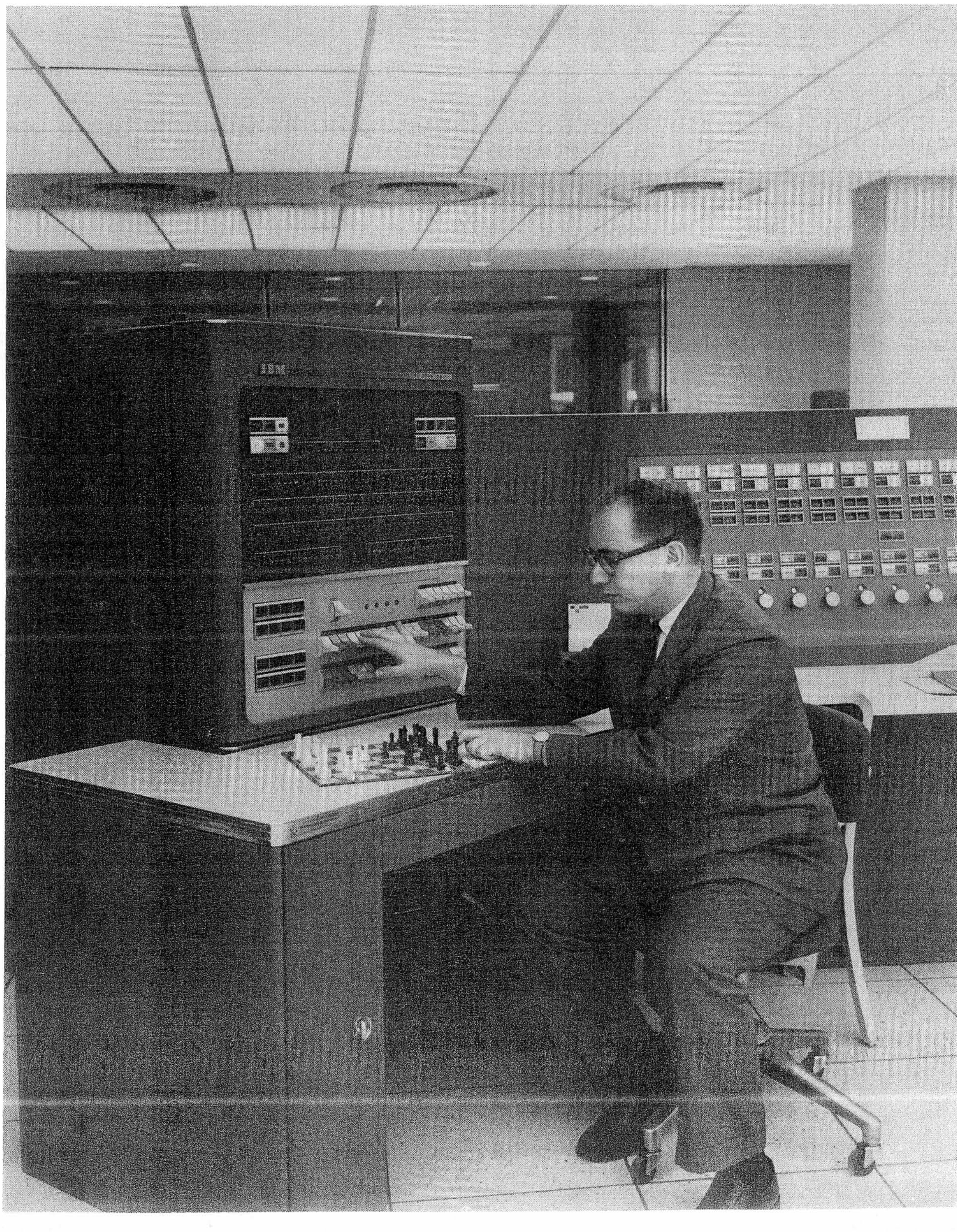
JULY
1958

IBM
704

(See page 195)

50 CENTS

Subscription Rate
ONE YEAR $5.50



CHESS REVIEW
THE PICTURE CHESS MAGAZINE

Vol. 26, No. 7
JULY, 1958

The World of Chess

INTERNATIONAL

European Team Competition

In a small team tournament in Paris, the French "A" team, comprising Muffang, Boutteville and Raizmann, finished on top, followed by Switzerland in second place.

A team event in Holland involving cities of various countries resulted in a victory by Rotterdam.

Matches in East Europe

Recent match scores were as follows: East Germany edged Poland by 13-12 in Section "A," while Poland turned the tables in Section "B" with a 10-5 tally. Sofia defeated Belgrade by 11½-8½. Big Yugoslav names on the first three boards — Trifunovich, Durasevich and Karaklaich — could only draw for Belgrade against Neikirch, Bobozov and Kolarov.

Correction

In listing the nine Zonal Tournaments and the players who have qualified from them for the coming Interzonal Tournament, CHESS REVIEW (June, p. 163) erred in designating Zone 7 as the "Latin American Zonal" and Zone 8 as the "Argentinian Championship and Zonal." The correct names are "Central American and Caribbean Zonal" for Zone 7 and "South American Zonal" for Zone 8.

UNITED STATES

NATIONAL

Amateur Crown

Dr. Eric W. Marchand of Rochester, New York outscored a huge field of 142 contestants to win the amateur championship sponsored by the United States Chess Federation. His 5½-½ tally was equaled by Stuart Margulies and Claude Hillinger, but Marchand was returned the winner on tie-breaking points. Lt. John Hudson, winner two years ago, lost only to Marchand, in the last round; and a large group crowded the 5-1 bracket.

REGIONAL

Creation of New Event

The initial Mid-Continent Chess Tournament, held in Russell, Kansas, attracted 36 players representing seven states. First prize was won by Dan Allen of Independence, Missouri, followed by runner-up Fred Tears of Dallas, Texas, and Claude Sponagle of Denver, Colorado, in third place. Mrs. Mable Burlingame of Phoenix, Arizona, was first in the Women's Division, and Howard Killough, Jr., of Russell, Kansas, topped the Junior Division. George Koltanowski functioned with his usual competence as tournament director. Foremost among organizers of the event were Dr. and Mrs. H. P. Killough, who did yeoman service prior to and during the three-day chess festival. So successful was the meet that the Russell Chamber of Commerce has already decided to sponsor a Second Mid-Continent Open in May, 1959. Excellent newspaper coverage was provided by the *Russell Daily News* and the *Russell Record*, both papers devoting much space to news accounts and photographs.

Bi-State Chess League

Teams from Racine and Kenosha in Wisconsin and from Waukegan and Des Plaines in Illinois, forming the Bi-State Chess League, held their first championship tourney in which Racine easily demonstrated superiority with a score of 6-0 in matches and 23-7 in games. Waukegan was second with 4-2 and 19-11. Top scorers for Racine were Domsky, Weider and Oberg.

CALIFORNIA

North and South clashed in the annual classic and staged their usual ding-dong battle, with the result in doubt up to the conclusion of the last few games. The final score was 32½-30½ in favor of the North, which now has fifteen victories to the South's seven in the entire series. Three matches have been tied. The South had scored two resurgent triumphs prior to the current setback, and was considered to have at least a fifty-fifty chance of making it three straight. As matters turned out, only a determined stand by the Northerners averted this outcome by a narrow margin. On the top five boards (the first-named players representing the North), I. Konig drew with I. Rivise; G. Ramirez won from H. Borochow; W. Pafnutieff lost to Sven Almgren; H. Gross won from Sonja Graf and E. Pruner drew with R. Brieger.

ON THE COVER

We have had several articles on chess-playing machines, and more have appeared elsewhere, with a good deal of debate on the possibilities. So, when Alex Bernstein, a mathematician in the Programming Research Department of I.B.M., had an article on chess-playing by the I.B.M. digital computer 704 in the Scientific American recently, we asked him to give an account which may clear up what the machine can and cannot do, for our readers.

For the most part, we shall let the account (on pages 208-9) speak for itself. It certainly should correct those readers who think, on the one hand, that a chess robot merely "memorizes" MCO and can be stymied by a move "out of (i.e., not in) the books."

On the other hand, the account does indicate that progress in chess-playing by machine has barely started. The limitations are very great. The major one, a running cost of hundreds of dollars an hour, is out of reach of the chess world. And it seriously limits the I.B.M. 704 in scope: to examine only 7 "plausible moves" to a depth of 4 "levels" (two turns to play by White and Black).

Nonetheless, the possibilities indicated are fascinating. If the monetary problem can be surmounted, more powerful machines, which can calculate more moves faster, may do better. With more elaborate "programming," the I.B.M. 704 can itself do better. At present, it only exchanges or retreats an attacked piece, whereas the choice of supporting it would add to the effectiveness of its play. It may add weighting of values for its criteria of mobility, area control and King safety, and it may also supplement the criteria in number.

Finally, the "self-improving" feature already in development may possibly prove able to do far more. It may for one small example determine a correct "weighting" for Bishop over Knight. And it may on a grand scale prove able to work out a Table of Criteria which can serve as a guide for truly competent chess-play.

A CHESS PLAYING PROGRAM FOR THE I. B. M. 704

By ALEX BERNSTEIN

Several articles have appeared in recent months on the subject of a chess playing machine. The I.B.M. 704 digital computer now has a program which enables it to play chess. It was written by the author and Michael DeV. Roberts of I.B.M. and H. A. Belsky and T. Arbuckle of the Service Bureau Corporation. The game which it plays is on an amateur level, and a good many of its moves are nothing to be ashamed of. It never leaves a piece **en prise**, a common failing of all chess players, and it can be seen that the level of its chess playing could be considerably improved were this program to be adapted for a bigger and faster machine.

This program was undertaken as a problem in simulation of human thinking. Most digital computer problems of today, such as payroll calculations, inventory control and prediction of missile and satellite flight paths present little difficulty in problem formulation, as methods exist and are easily stated for solving these problems. The equations can be written, and a procedure stated which, when applied to the specific problem, will invariably give rise to the proper result. This circumstance is not true in respect to chess. No one has yet specified a set of formulas such that following them will enable a person to arrive at the best, or even a reasonably good move in chess. This, indeed, is one of the fascinations of the game. Therefore, the interest in this problem for the computer. The machine must be made to select moves and evaluate positions much as a human being does; for examining in depth, that is to say several moves ahead for all possible moves, would still require far too much time.

The program works in the following fashion. The board position is scanned, square by square, and tables of information are generated. These tables contain all immediate information, including a list of moves, which pieces are attacked, and by what, which pieces are defended, and by what, as well as information relating to doubled pieces (Rooks behind Rooks, etc.).

A score is also evaluated for this position, based upon mobility, area control, King defense, and material. Each of these criteria is weighted; and, at present, mobility, area control and King defense (number of controlled squares around the King) are set equal to each other. Material is set as worth more in order to discourage the program from sacrificing a piece, say, for a few points of mobility or area control. The squares themselves are also currently set as equal in importance, but different values may be assigned to the control squares, for example, or the ones surrounding the Kings.

In order to avoid examining the consequences of all possible moves, since most of them are bad, anyway, and since it would take far too much time, a set of decision routines were written which select a small number (not greater than seven) of strategically good moves. These moves are now referred to the Tree, the part of the program which examines in depth. The Tree operates thus: it makes the first move of this set of seven plausible moves and asks the decision routines for seven plausible replies. Having received them, it executes the first of these and again asks for seven responses, performs the first and once more asks for seven counter responses. It executes the first one of these counter responses, scores the position, undoes the move, executes the second, scores the position, and so on, until it has gone through all seven. It takes the highest score and brings it back as the score of the first move on the third level, for it must assume that, if it made that first move on the third level, the opponent would choose his best reply. It then undoes its first move on the third level, executes the second move and descends once more to the fourth level. In this fashion, it examines 2800 different positions and arrives at that move which, in its estimation, will leave the opponent with the worst possible position.*

The generation of the Plausible Move Table is performed by the decision making section of the program. This section asks a series of questions concerning the state of the board and, depending on the answers to these questions, manufactures moves. The questions are:

1. Is the King in check?
2. a. Can material be gained?
   b. Can material be lost?
   c. Can material be exchanged?
3. Is castling possible?
4. Can minor pieces be developed?
5. Can squares defended by the Pawns of a Pawn chain be occupied?
6. Can open files be occupied or invaded?
7. Can Pawns be moved?
8. Can pieces be moved?

If the answer to (1) is yes, King moves, interpositions and the possible capture of the checking piece are entered in the Plausible Move Table, whereupon control returns to the Tree. If the answer to (1) is no, the decision routines go to each successive question and construct whatever moves exist which satisfy the questions. When the Plausible Move Table reaches seven, control is returned to the Tree.

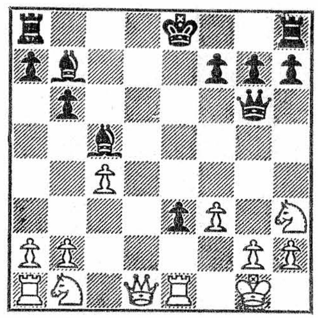
* That is, in any given position, IBM 704 considers 7 plausible moves, 1st level, 7 positions; 7 replies to each, 2d level, 56 positions; 7 responses, 3d level, 399 positions; 7 counter-responses, 4th level, 2800 positions. — Ed.

These decision routines work for either side of the board, allowing the machine to generate moves both for itself and its opponent. It will be noted that, at the beginning of the game, only questions (4) and (7) are answered affirmatively and, therefore, the Plausible Move Table will have in it the moves: N–QR3, N–QB3, N–KR3, N–KB3, P–K4, P–K3, P–Q4. As the game progresses, exchanges, key squares and open files will take on added importance, while, eventually, Pawn moves will take on repeated importance in the end-game.

Were the machine to have a larger memory, more questions could be asked: for example, are there doubled Pawns, passed Pawns, isolated Pawns, Bishops of opposite colors, and so on, and each of these situations would give rise to different moves. There is no reason why a machine with a sufficiently complicated set of decision routines could not play very well; and, in the not too distant future, we may see other chess programs performing. Newell, Simon and Shaw, and also McCarthy are currently working on other chess playing programs.

Here is a sample game.

IBM 704 White	Opponent Black		
1 P–K4	P–K4	9 B–N5†	N–B3
2 B–B4	P–QN3	10 P–B4	PxP
3 P–Q3	N–KB3	11 BxN†	QxB
4 B–KN5	B–N2	12 PxP	P–K5
5 BxN	QxB	13 N–N5	Q–N3
6 N–KB3	P–B3	14 N–KR3	P–K6
7 O–O	P–Q4	15 P–B3	B–B4
8 PxP	PxP	16 R–K1	. . . .



(The preceding moves are not recommended: the IBM 704 sheds Bishops for Knights with unjustified nonchalance, for example, and 15 N–B4 appears more effective a defense than its 15 P–B3, but its last move is quite good. — Ed.)

16	O–O
17 N–B3	. . . .

(This move, however, spells ruin, a curious oversight when 17 Q–K2 serves so well. — Ed.)

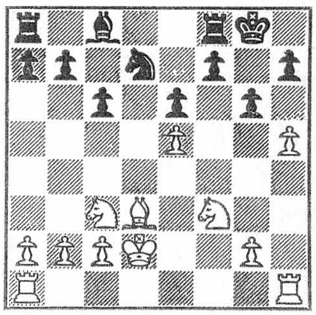
17	P–K7§	20 N/3xQ	Q–B7
18 N–B2	BxP	21 P–N3	QR–Q1
19 P–KN3	PxQ(Q)	22 P–KR4	RxN
		Resigns	

The main weakness of the program is that it is programmed to move away attacked pieces rather than to defend them. The latter could be done at the cost of lengthening the time required to make a move, which is now on the average of eight minutes. The program is capable of looking ahead more than four half moves, but the time increases fifty fold with each two half levels and is, therefore, impractical.

It is interesting to note that a slight change in the program, namely, having it look at all possible moves, would make it capable of solving two move mate problems in a very reasonable time, probably under fifteen minutes.

Here is a game which the computer was well on its way to winning but had to be stopped as the time for this experimental project ran out.

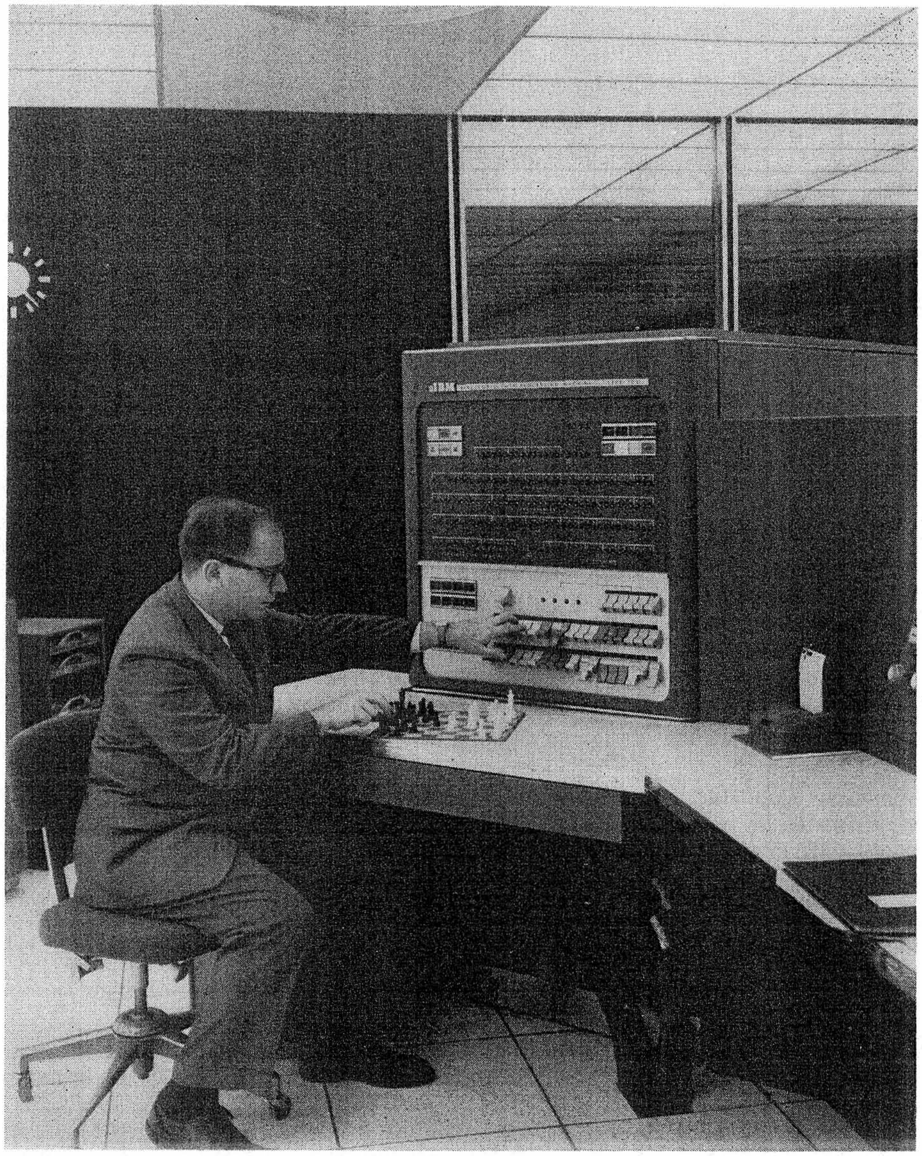
IBM 704			Beginner
1 P–K4	P–K4	9 P–KB4	B–R3
2 B–N5	N–KB3	10 PxP	BxB†
3 P–Q4	N–KN5	11 KxB	QxP/4
4 QxN	P–QB3	12 N–KB3	Q–K3
5 B–B4	Q–R4†	13 QxQ†	QPxP
6 B–Q2	Q–R5	14 P–K5	O–O
7 B–Q3	QxQP	15 P–KR4	N–Q2
8 N–QB3	P–KN3	16 P–R5	



(Comment on an incomplete game is not apt to be illuminating; but it is apparent that a machine-like infallibility as to leaving piece **en prise** is an asset which puts IBM 704 at better than sub-amateur level; it seems that IBM 704 can press an attack with material superiority; yet it also seems that IBM 704 is almost careless in respect to exchanging or permitting exchanges too readily — that is, before being materially ahead. — **Ed.**)

At present, the machine will make the same move whenever it encounters the same position. This situation, however, can be remedied by adding a self-improvement feature.

There are two different routines which could be added to our chess program. The first would work in this manner: assume the machine loses a game; it will then examine the opponent's moves to find those moves which the opponent made which the machine did not predict. If the move which the opponent made was, however, in the set of seven plausible moves, then a different evaluation must exist which would arrive at this move instead of the one at which the program did arrive. The self-improvement routine would change the weights of the criteria until this move would give the highest score.



Alex Bernstein at the control of the I.B.M. 704

Another scheme would be this: again assume that the machine loses a game and, upon examination of the opponent's moves, it finds that one of these was not only not predicted but was not even in the set of plausible moves; this routine would reconstruct the situation and go through the decision routines again, not stopping this time when it had generated seven moves but continuing until the move which the opponent actually made was generated. When that move was generated, this routine would determine which question generated this move, and the ordering of the questions might be changed to produce that missing move.

The question arises: why would we want to change the weights of the criteria or the ordering of the decision routines? Because we have no real certainty that these weights or this ordering is really valid. They were not arbitrarily chosen; but, although they were based on our knowledge of chess, they are by no means infallible. Who today knows whether the central squares are worth twice as much as the outside squares, three times or just one and a half as much? We cannot tell whether the Bishop should be given a higher value than the Knight or whether its greater mobility and area control is sufficient to make it more valuable to the mechanistic evaluation of the program.

One could even get the program to take advantage of strategic mistakes made by the opponent. If the program were to examine the opponent's moves, which again are outside the set of seven plausible moves and found that it had a low evaluation, it could be programmed to determine which strategic principle the opponent's move violated, much in the same way as it generates plausible moves. It could then store this information for future use as to the sort of weakness which the opponent might have and to what sort of attacks it might succumb. This procedure can, of course, lead to real complexity, for one might try to fool the program with a weak move, but it will also pose very interesting problems which might gain us useful information.