CMU-CS-71-109

THE TECHNOLOGY CHESS PROGRAM

James J. Gillogly

29 November 1971

Carnegie-Mellon University
Department of Computer Science
Pittsburgh, PA 15213

This work was supported in part by a RAND Corporation Fellowship and in part by the Advanced Research Projects Agency of the Office of the Secretary of Defense (F44620-70-C-0107) and is monitored by the Air Force Office of Scientific Research. This document has been approved for public release and sale; its distribution is unlimited.

ABSTRACT

A chess program has been developed which plays good chess (for a program) using a very simple structure. It is based on a brute force search of the move tree with no forward pruning, using material as the only terminal evaluation function, and using a limited positional analysis at the top level for a tiebreak between moves which are materially equal. Because of the transparent structure, this program is proposed as a technological benchmark for chess programs which will continue to improve as computer technology increases.

Acknowledgements

I am indebted to Hans Berliner, World Correspondence Chess Champion, who developed the elegant techniques used in the positional analysis, and whose patient discussions helped to clarify many of the conceptual problems. The basic idea of the Technology Program is due in large part to Allen Newell.

Chess and Technology

Until recently the main effort in chess programming has been to develop programs which selectively (and hopefully "intelligently") examine a small subset of the legal moves in any position. The surprising performance of the Varian minicomputer (programmed by K. King and C. Daly) in the First Annual Computer Chess Championship (New York 1970), although due primarily to good luck in the pairings, led to increased speculation about the possibility of playing respectable chess with an unselective "brute force" program.

We were led to reconsider again programs that would simply generate all legal moves to a fixed depth, then evaluate the final position only with respect to material. Such programs would be very small and would have a transparent control structure, so it would be easy to reprogram them for faster computers as they become available. This type of program would need about a factor of 7 in computing speed to go each additional ply deeper in the same time. Since this speed

Increase Is not unreasonable for (say) a decade of computer technology/ a "technology program" would provide an Increasing baseline for chess programming against which more sophisticated programs could be compared; that Is, In order to Justify the effort of bringing up a complex program, that program must be able to beat a technology program.

Initial experiments Indicated that this baste design does not produce a useful baseline, since the standard of play Is low for any reasonable depth of search. In practice, the problem Is that before anything tactical (i.e. discoverable by changes In material) happens, the program has a hopeless positional disadvantage In terms of future opportunities. This type of program also makes tactical blunders due to evaluating at non-quiescent positions, although this effect decreases with Increasing depth. (An example of the play of this type of program Is given In the appendix, game *k.* White Is played by a technology-type program which uses no positional analysis but which has a limited knowledge of quiescence, and Black Is played by the full program described In this paper.)

Consequently a more useful definition of a Technology program was sought. To be a useful benchmark the program should take only a few man-months to Implement; therefor It must have a simple structure. The search should depend primarily on brute force, rather than on chess-speclfIc heuristics; hence It probably could not afford forward pruning, which requires substantial analysis at each position. A reasonable limitation might be 5% or less CPU time spent on

chess-specific heuristics.

A program, TECH, has been developed which conforms to this less restrictive definition of a technology program. TECH was conceived in October 1970 and played its first complete game in November 1970. In March 1971 TECH beat a 1968 version of the Greenblatt program(*) with White and drew with Black. TECH joined the U. S. Chess Federation in May 1971, and has since played 21 USCF rated games (see Table 1). In August 1971 TECH won the second place trophy in the ACM-sponsored Second Annual Computer Chess Championship , placing behind Chess 3.5 (the defending champion, written by Slate, Atkin, and Gorlen) and ahead of six other programs.

TECH is written in BLISS(**), a system implementation language developed at Carnegie-Mellon University. BLISS was chosen because (1) the language was designed to yield efficient object code, and (2) it is a higher-level language, and thus more legible than assembly language. TECH currently runs ona PDP-10.

(*) Several versions of the Greenblatt program are available through DEC and others. The version used is somewhat identified by the fact that teletype input is in line mode and the prompting character is "*". the tournament setting was used: SETW 15 15 9 9 7.

(**) Wulf, W. A., et al., BLISS Reference Manual, Pittsburgh: Computer Science Department, Carnegie-Mellon University, 7 Apr 1971.

This paper is an attempt to describe a technology program (TECH) in sufficient detail for others to build a similar program and compare its performance with TECH's. Several sample games are appended to demonstrate the level of play that can be achieved.

Design

The move production mechanism consists of two main parts: positional and tactical analysis. The portions of the tree affected by these components are shown in Figure 1. The positional analysis routine pre-sorts the moves at the top level (ply=1) so that the move which has the best superficial positional score is considered first. No tactical considerations are included in the positional analysis. The tactical analyzer is a brute force tree search which investigates all moves to a fixed depth, applies a simple quiescence scheme, then evaluates the final position using material as its only criterion. Alpha-beta will accept the first of a group of materially equal best alternatives; the positional pre-sort ensures that the first of these alternatives will also have the highest superficial positional value.

In addition to invoking the move production routine, the supervisor controls utilization of the time when the opponent is planning his move (TECH is the only program so far which does this). TECH uses a 2 ply search to guess its opponent's move, then begins calculating its move on the basis of that assumption. If the opponent

Table 1 : TECH's USCF-rated events

Date	Event	Rounds	Points
May 1971	Golden Triangle Open	5	1.5
June 1971	Fred Thompson Memorial	5	2.0
Fall 1971	Walled Knights Open	4	1.5
Sept 1971	Pittsburgh Industrial League	2	1.0
Oct 1971	Gateway Open	5	2.0

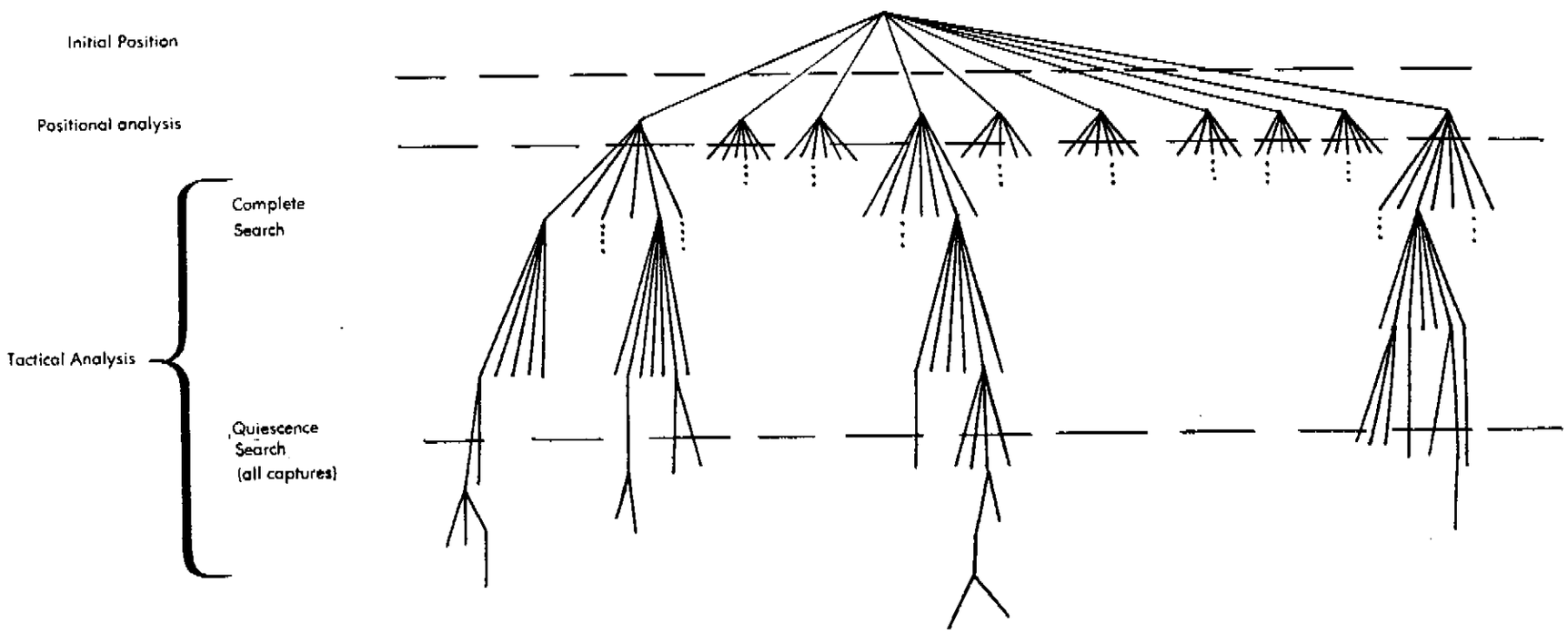

Figure 1: Move production for a 3 ply search with quiescence

makes the expected move, TECH will print its reply immediately if finished, or continue the computations. The supervisor also gathers statistics on timing and on the size and shape of the tree, and controls the interface between the program and the operator, e.g. by recognizing and printing standard English notation, considering draw offers, setting up problem positions, and saving the current program state.

Tactical Analysis

The heart of the Technology Program is the brute force search, or tactical analysis component. All moves are searched to a fixed depth (usually five ply in the middle game), after which quiescence is approximated by investigating all sequences of captures. Even using alpha-beta, this search strategy can result in as many as 500,000 bottom positions in a move analysis tree under tournament conditions, which means that TECH looks at many more positions than other programs in a given amount of time. By way of comparison, the Raymond/Ridley program which took third place at the Second Annual Computer Chess Championship looked at less than 100 bottom positions on each move.

This speed is achieved by using a simple terminal evaluation function and efficient move generation. Most chess programs use a terminal evaluation function which includes material, king safety, pawn structure, development, and many other terms; TECH's terminal

evaluation function Is simply the value of TECH's material minus the onoonent's material.

Potential legal moves *re generated in the usual way/ I.e. by adding offsets to a piece's location on the board/ which Is represented as a 120-word vector (see Fig. 2). For example/ the potentially legal moves for a knight on White's QR3 ar* obtained by adding the offsets 8/ 12/ 19/ 21/ -8/ -12/ -19 and -21 to square ***kl*** and testing the contents of those cells. If any of the resulting moves Is Illegal (moving Into check/ moving a pinned niece) they will be eliminated by the tree search when It notices that a king can be captured. Since this Is a relatively rare occurrence/ a considerable saving of time Is achieved by not checking absolute legality of potentially legal moves.

The tree Is represented as a stack of single words which contain the Information necessary to travel up and down the tree (see Fig. 3). Two types of positions are represented by each word In the tree/ distinguished by bit 35: those positions whose successors have been Investigated/ and those whose successors have not. (see Fig. ***k)*** The stack Is Initialized by pushing the positionally sorted top-level moves onto the tree so that the best move will be popped first. This move Is marked In bit 35/ the move Is executed/ and Its successors are generated and pushed onto the tree. When a bottom position Is reached/ It Is evaluated and the value Is backed up by mlnlmaxlng until the next unevaluated position Is reached. The successors of that node are then evaluated.

	110	111	112	113	114	115	116	117	118	119	
110	7	7	7	7	7	7	7	7	7	7	119
100	7	7	7	7	7	7	7	7	7	7	109
90	7	-4	-2	-3	-5	-6	-3	-2	-4	7	99
80	7	-1	-1	-1	-1	-1	-1	-1	-1	7	89
70	7	0	0	0	0	0	0	0	0	7	79
60	7	0	0	0	0	0	0	0	0	7	69
50	7	0	0	0	0	0	0	0	0	7	59
40	7	0	0	0	0	0	0	0	0	7	49
30	7	1	1	1	1	1	1	1	1	7	39
20	7	4	2	3	5	6	3	2	4	7	29
10	7	7	7	7	7	7	7	7	7	7	19
0	7	7	7	7	7	7	7	7	7	7	9
	0	1	2	3	4	5	6	7	8	9	

Figure 2: Initial board representation

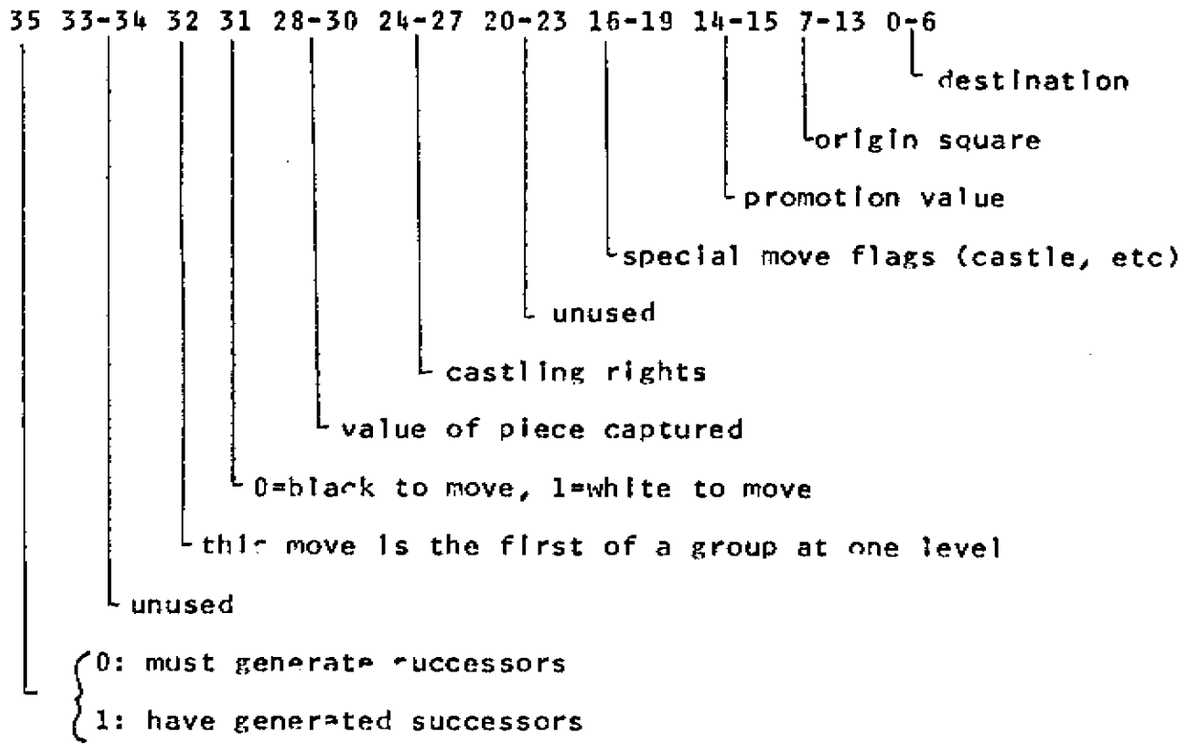

Figure 3: Tree word bit allocations

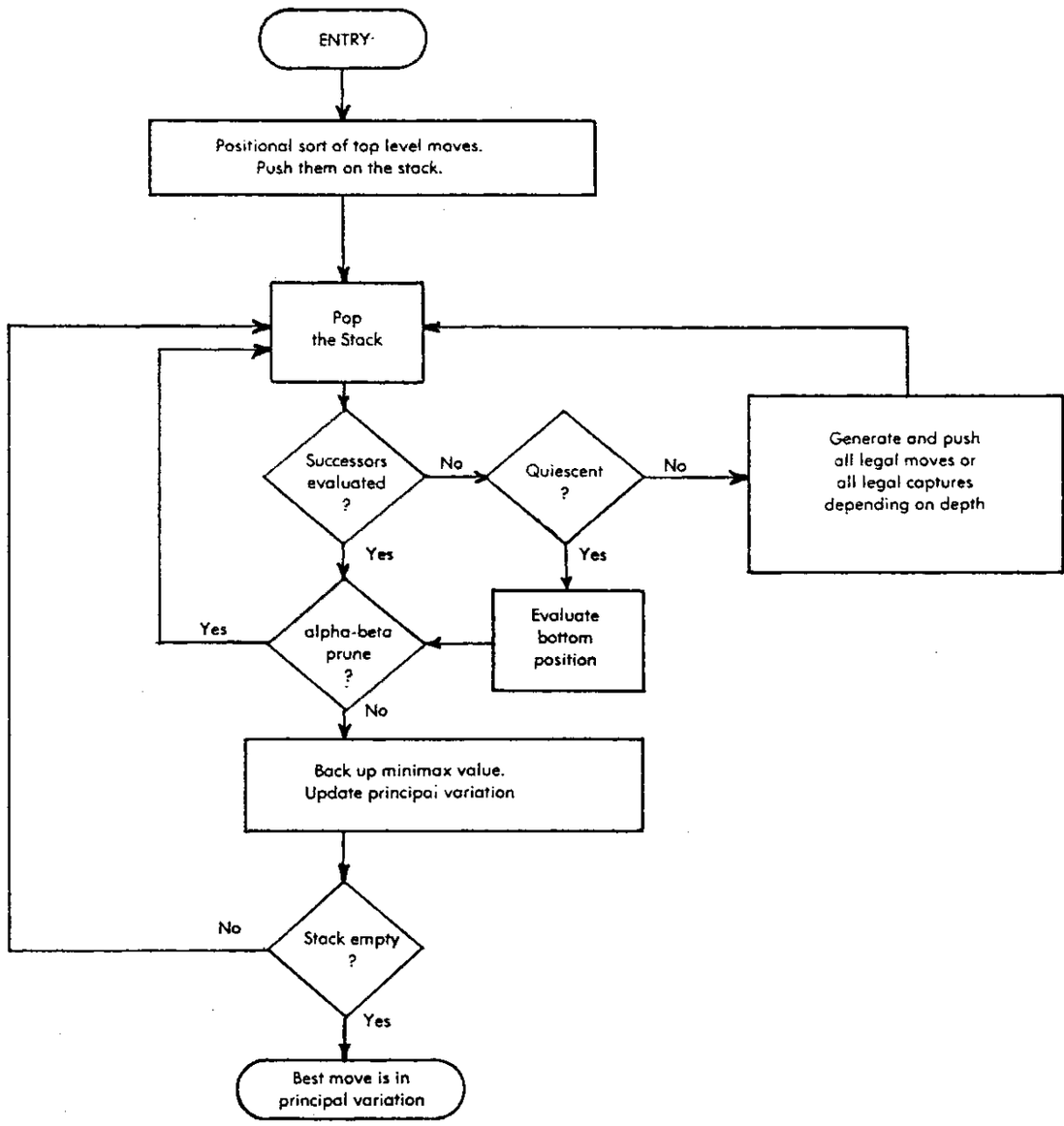

Figure 4: Tree Discipline

For the sake of efficiency moves are retracted while traversing the tree, rather than recopying the previous position at each point. The value of material is incrementally updated only when a capture is encountered, so that the terminal evaluation function consisists of the single BLISS assignment statement "AB(.PLY)←.MATERIAL;". (AB is a vector which contains the best value so far at each level. AB stands for Alpha-Beta.) The basic piece values are P=100, N=B=330, R=500, Q=900, and K=15000. These values can be modified for one side by the positional analysis.

The efficiency of alpha-beta is greatly increased if the best moves are considered first(*). Since the refutation of a bad move is often a capture, all captures are considered first in the tree, starting with the highest valued piece captured. This is an inexpensive process, since captures can be recognized and sorted during move generation. The killer heuristic is also used: if a move is a refutation for one line, it may also refute another line, so it should be considered first if it appears in the list of legal moves.

(*) Slagle, J. R. and Dixon, J. K. "Experiments with Some Programs that search Game Trees," Journal of the ACM, Vol. 16, No 2, Apr 1969, pp. 189-207.

Positional Analysis

The soul of the Technology Program is in the positional presorting routine. This contains nearly all the chess-specific heuristics used by TECH. When used with the tactical search, it almost always can achieve a strong opening, even against players and programs which use a book. Although TECH does not use a book of openings, it follows standard opening play very closely (see Appendix, game C). TECH recognizes five phases: opening, middle game, engame with pawns, general endgame, and endames where TECH must mate with pieces only. Remember that all the heuristics described in this section are applied only to the moves at the top level of the tree.

In each phase TECH will readjust the piece values given in the previous section if necessary. Specifically, if TECH is ahead by 200 points (2 pawns) or more, the new value of a piece (not a pawn) is computed by the formula

new value ← oldvalue * max (.6, opponent's material/TECH's material).

This encourages TECH to exchange pieces when ahead by making its pieces worth as little as 60% of its opponent's pieces.

After the opening phase TECH will also adjust its maximum depth for the next tactical analysis by comparing the average amount of time per move available before the next time control with the average amount of time used on the last five moves. If it has taken too much time, it decreases the depth. If it has taken enough time less than

```
0  1  2  3  3  2  1  0
1  3  4  5  5  4  3  1
2  4  6  7  7  6  4  2
3  5  7  8  8  7  5  3
3  5  7  8  8  7  5  3
2  4  6  7  7  6  4  2
1  3  4  5  5  4  3  1
0  1  2  3  3  2  1  0
```

Figure 5: Center control array

```
2  2  2  2  2  2  2
2  8  8  8  8  8  2
2  8 10 10 10  8  2
2  8 10  K 10  8  2
2  8 10 10 10  8  2
2  8  8  8  8  8  2
2  2  2  2  2  2  2
```

Figure 6: Middle game king field

```
8 12  P  12  8
3  5 4/8  5  3
4  6 5/7  6  4
1  2 1/5  2  1
    0/1
own pawn/opponent's pawn
```

Figure 7 : Pawn endgame passed pawn field

the allotted amount it increases the depth. It needs a factor of 3 to go up from an even depth to an odd one, and a factor of about 7 to go from an odd depth to an even one.

1. Opening

The opening is defined to be the first eight moves. The most important heuristic in the opening evaluation is occupation of the center. Each square on the board is weighted with a desirability value ranging from 0 points for the corners to 8 points for the center (Fig. 5). Each move represents a net gain or loss of centrality. For example, N-KB3 would yield a gain of 5 points in centrality. This is multiplied by the priority factor for the piece to move: P=1, N=4, B=3, R=2, Q=1, and K=-1. Thus N-KB3 would have a final score of 20 points for centrality. Notice that the king is encouraged to move away from the center in the opening, since its center-tropism factor is negative. This heuristic alone dictates a very reasonable opening with rapid development.

Each move is given a final positional score of the centrality term plus the value of each of the following heuristics which applies to it:

Pawn from K2 to *Kk* 30 points
Pawn from K3 to *Kk* 2 points
Pawn from Q2 to 20
Pawn from Q3 to *Qk* 2
0-0 : 30
0-0-0 : 10
N-R3 : -15
Piece to K3 or Q3 blocking a pawn : -50
Piece moving from king side : 2
Playing Petroff defence : -50
Capture with pawn toward center : 5
Capture with pawn away from center : -5
Pawn capture leading to multlpled isolated pawns : -10
Wing pawn advance : -10
Capture unsupported center pawn : 50
Capture supported center pawn : -15

2. Middle Game

The middle game begins with the ninth move and continues until one side has less than 1950 points worth of material, excluding the king (each side has *kk20* In the Initial position). The center control heuristic Is still used, but the priority factors are slightly altered: P»3, N«i*, B»3, R*2, Q«l, and K«l. Since most pieces have found their best squares by the middle game, this factor has less Influence than In the opening. Each move Is credited with a mobility term, which Is the number of potentially legal moves available after the move Is made. Movement of a piece Into the opponent's king field (see Fig. 6) Is rewarded In the same way as the center control heuristic, and the net gain Is again multiplied by the priority for that piece. This heuristic occasionally results In a king-side attack.

The pawn heuristics are the same as In the opening, except that advances of wing pawns get -5 instead of -10. Castling values are the

same as in the opening. If TECH is ahead in material, piece captures get 10 points more. Moving a piece which blocks the KBP or QBP is rewarded with 5 points.

3. Endgame with Pawns

The most important goals in pawn endgames are advancing and blocking passed pawns. Each move is credited with the net gain in the passed pawn field shown in Figure 7. This allows TECH to escort the pawn (if its own) or block it (if the opponent's). The king field (Fig. 8) and center control arrays are used only for king moves.

Pawn moves are weighted by the rank of their destination and by whether they are opposed:

Rank	Opposed	Unopposed
3	2	3
4	1	5
5	3	10
6	4	13
7	-	23
8	-	80

If TECH has multipled pawns on a file, only the first is given this bonus; the other pawns lose 10 points.

4. General Endgame

As in the pawn endgame, TECH's main goal is to promote. The pawns are given the same weights for advancing as in the preceding section. The material value of a pawn is raised from 100 to 120; if TECH has 2 or less pawns, they are worth 190 each. A move which

1	1	2	3	2	1	1
1	3	4	5	4	3	1
2	4	6	6	6	4	2
3	5	6	K	6	5	3
2	4	6	6	6	4	2
1	3	4	5	4	3	1
1	1	2	3	2	1	1

Figure 8: Pawn endgame king field

4	4	5	6	5	4	4
4	8	10	10	10	8	4
5	10	10	10	10	10	5
6	10	10	K	10	10	6
5	10	10	10	10	10	5
4	8	10	10	10	8	4
4	4	5	6	5	4	4

Figure 9: General endgame king field

4	4	5	6	5	4	4
4	8	10	9	10	8	4
5	10	10	10	10	10	5
6	9	10	K	10	9	6
5	10	10	10	10	10	5
4	8	10	9	10	8	4
4	4	5	6	5	4	4

Figure 10: Piece endgame king field

places a rook behind a passed pawn of either color is rewarded with 15 points. The center control term uses priorities of P=0, N=4, B=3, R=1, Q=1, and K=4. This encourages the king to centralize. TECH also uses the king field mask (Fig. 9) to minimize the distance between kings. As in the middle game, the mobility is added to the score for a move.

5. Endgame with Pieces

Unlike the other forms of endgame, TECH's goal in the endgame with pieces is to drive its opponent's king to the edge in order to deliver mate. This is achieved by doing a small (2 ply) tree search and using as an evaluation function:

(1) -32* opponent's king location on the center control field (Fig. 5)

(2) 2* opponent's king location in TECH's king field (Fig. 10)

(3) TECH's king location on the center control field, and

(4) the sum of TECH's piece locations in TECH's king field divided by the number of TECH's pieces (to keep pieces near the king as a tiebreak).

This method of forcing the king to the side of the board is due in part to Slate, Atkin, and Gorlen (authors of Chess 3.5).

Statistics

In order to assess present and future implementations of technology programs, it is necessary to determine how much effort is required to search one ply deeper, and how much better an n+1-ply program is than an n-ply program. Statistics are presented in this section to shed light on the former question, but the latter question has not yet been settled. To get the statistics, every fifth move of each of TECH's tournament games was analyzed at depths of 2, 3, 4, and 5 ply.

One measure of search effort used frequently in the literature is the number of bottom positions (NBP) in the tree(*), shown in Figure 11. As Slagle and Dixon's results indicate, the factor required to go from even to odd ply (about 8 in this sample) is larger than that required to go from odd to even (about 4). However, NBP is not a good measure of effort for TECH, since the amount of time spent processing bottom positions is negligable compared to the cost of move generation to arrive at those bottom positions. The CPU time (Figure 12) and the number of move generations (Figure 13) each show less effort to go from even to odd ply (about 3.5) than from odd to even (about 7). All the CPU times are inflated by about 10% due to the overhead of gathering statistics.

(*) Slagle, J. R. and Dixon, J. K., op. cit.

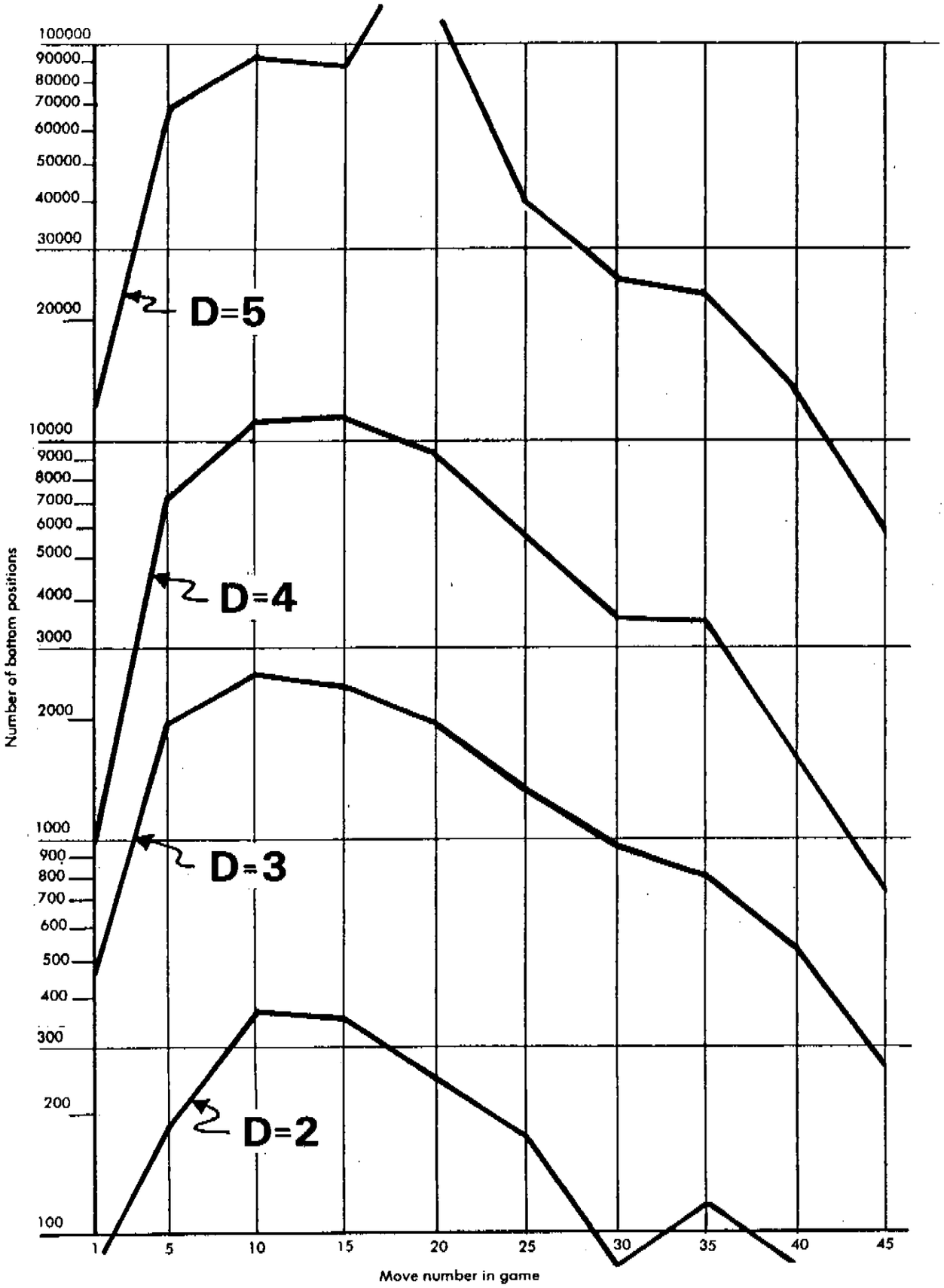

Figure 11: Number of bottom positions vs. move number

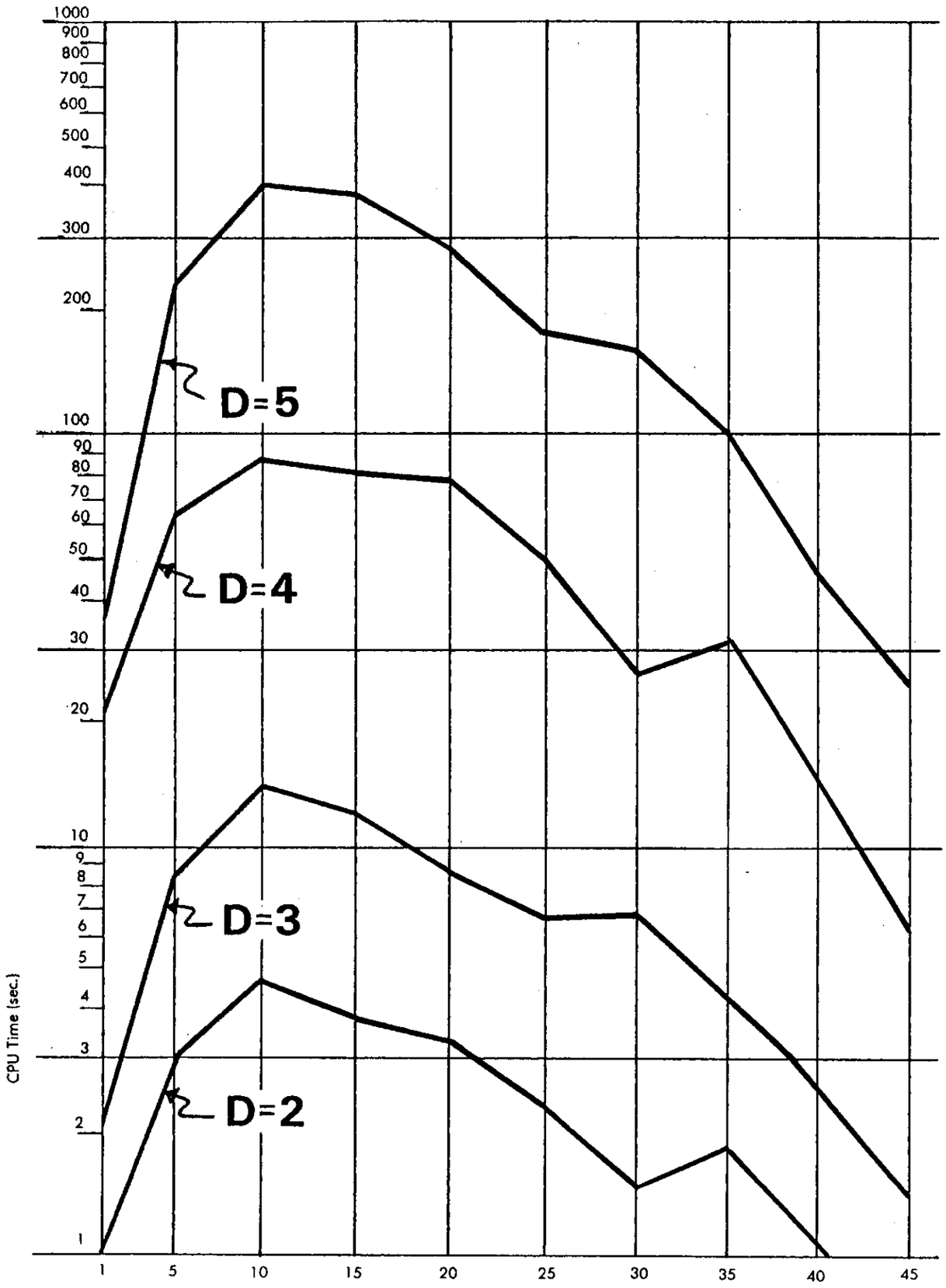

Figure 12: CPU time vs. move number

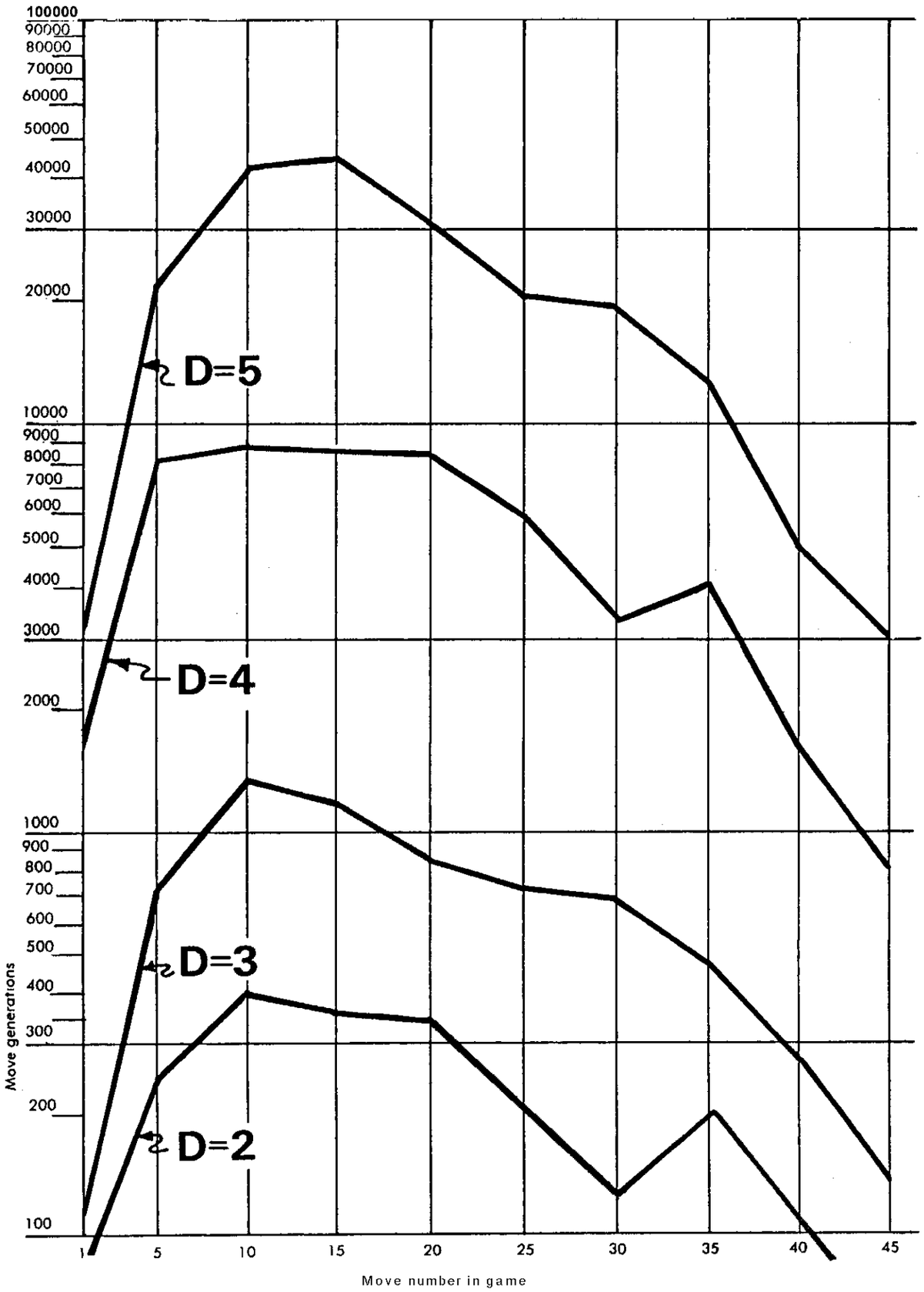

Figure 13: Number of move generations vs. move number

The following figures will be useful in comparing different technology programs:

PDP-10 cycle speed: 1 microsecond

TECH legal move generation

(all potentially legal moves): 8000 microseconds

Board updating (making or retracting moves) : 1300 microseconds

TECH spends its time as follows:

Legal move generation: 50%

Tree management : 28%

Board updating : 17%

Positional analysis : 1%

Everything else : 4%

The Future of Technology Programming

The experience gained from TECH suggests other types of programs within the "Technology" framework. For example, it would be possible to expand greatly the positional module, specifically including the basic endgames and allowing a deeper positional search. Some effects of the positional search could also be included directly in the tactical search, so that without additional computation the program could sacrifice a small amount of material for significant positional advantages.

TECH is now close to its final form. Some of the planned minor

modifications are implementation of Greenblatt's secondary search scheme (*), variable individual piece values (to distinguish between "good bishops" and "bad bishops", for example), and a method of doing some incremental positional updating within the tree. A major goal is to get a firm USCF rating for the current configuration, so the technology baseline would be established. TECH's current rating is 1147 based on its first tournament.

The future of chess programming lies with programs that (unlike TECH) have some understanding of chess, and which do a restricted, goal-oriented search. TECH's main limitations (no planning ability, no deep positional knowledge) seem to be inherent in the concept of a technology program. However, the goal of TECH is to be a useful benchmark for chess programming, rather than to become the world's best chess player. In this respect TECH seems to quite successful.

(*) Greenblatt, R., Eastlake, D. and Crocker, S. "The Greenblatt Chess Program, Proc. AFIPS FJCC 1967, pp. 801-810.

Appendix

Annotations by Hans Berliner

Game 1: COKO III vs TECH, Second Computer Chess Championship

	COKO III	TECH		COKO III	TECH
1.	P-K4	P-K4	21.	R-QB5	Q-K3
2.	N-KB3	N-QB3	22.	B-QB1	P-QB3
3.	B-QB4	N-KB3	23.	P-Q4	R(QR1)-K1??(H)
4.	P-Q3 (A)	P-Q4	24.	RXP(K5)	Q-KN3
5.	BXP? (B)	NXB	25.	RXR	QXR
6.	PXN	QXP	26.	Q-KB2	Q-K3
7.	N-QB3	B-QN5	27.	Q-B1	R-B4
8.	O-O	BXN	28.	P-KR4	P-B4! (I)
9.	PXB	O-O	29.	P-Q5	Q-Q3
10.	N-KN5	B-KB4	30.	Q-KR3	Q-K4
11.	R-QN1	P-KB3 (C)	31.	Q-KB1	QXBP
12.	P-QB4	Q-QB4	32.	P-Q6	Q-Q5! (J)
13.	N-KR3?? (D)	BXN (E)	33.	Q-K2	QXQP
14.	B-K3	N-Q5!	34.	Q-K8 CH	R-KB1
15.	PXB	Q-QB3!! (F)	35.	Q-QR4	R-KB4
16.	P-QB3?	N-B6 CH	36.	Q-K8 CH	R-KB1
17.	K-R1	N-Q7 CH!	37.	Q-QR4	Q-K3 (K)
18.	P-KB3	NXR/KB1	38.	Q-QN3	Q-K7!! (L)
19.	QXN	P-KB4 (G)	39.	P-KR3	R-Q1 (M)
20.	R-QN5	P-KB5	40.	BXP	R-Q8 CH
				resigns	

(A) A passive move which gives up any hope of advantage.

(B) PXP was necessary. Now Black obtains a dominant position.

(C) Black has attained a fine position by making only simple classical developing moves.

(D) White should continue with B-K3 and then bring the N back to KB3. The text results in a terrible weakening.

(E) A stroke of fortune. TECH does not recognize doubled P's in its evaluating function, but selects this move because of K proximity.

(F) Again serendipity. This is computed to be the best square for the Q (It is necessary to move it to avoid material loss due to P-QB3) and the threats on the diagonal are not part of the evaluation. Now BXN

or P-KB3 had to be played.

(G) Correctly pursuing a policy of gaining space, but on the next move P-K5 is better.

(H) This loss of a P was later found to be the result of a program bug, and makes things difficult. Simply PXP wins.

(I) It is interesting that this important strategic break comes as a result of the positional heuristics, since there is no material gain involved.

(J) Now White is clearly lost. It is only a matter of time.

(K) The position repetition mechanism asserts itself in order to replace the favored R-B4 by a previously untried move.

(L) A truly great move. Instead of the "obvious" (to humans) Q-K8ch which wins the B, TECH recognizes the mate in 1 possibility and prefers Q-K7 to Q-K8ch, since it is more centralizing and the threat to win the B does not go away.

(M) Now TECH sees it can win the Q and does so, which leaves it in a situation that it can readily convert to mate. Actually a mate was possible by 39. ... Q-B8ch, 40. K-R2 R-K1 threatening both R-K7 mate and Q-B7ch followed by R-K8 mate, both of which cannot be answered; but this is outside TECH's horizon.

Game 2: TECH (depth 4) vs David Levy, International Master
Played 29 July 1971 at Carnegie-Mellon University.

	TECH	Levy		TECH	Levy
1.	P-K4	P-Q3	21.	R-K3	N-Q5
2.	P-Q4	N-KB3	22.	N/2-K4	R/B1-Q1
3.	N-QB3	P-KN3	23.	K-KR1 (D)	BXN (E)
4.	N-KB3	B-KN2	24.	RXB	RXQP
5.	B-Q3	O-O	25.	RXP	R-Q3
6.	O-O	P-QB3	26.	N-K4 (F)	QXQ
7.	B-K3	QN-Q2	27.	NXQ	NXP
8.	Q-Q2	P-QN4	28.	NXP	R-Q5 (G)
9.	B-KR6	N-QN3	29.	N-K5	RXBP
10.	BXB	KXB	30.	RXRP	RXP!
11.	N-KN5	N-QB5? (A)	31.	RXP CH?? (H)	RXR
12.	BXN	PXB	32.	R-Q1	N-K6
13.	P-KB4!	R-QN1	33.	R-QB1	R/B2-B7
14.	P-K5! (B)	N-K1	34.	N-Q3	R/N7-B7
15.	PXP? (C)	NXP!	35.	R-R1	RXNP
16.	QR-N1	B-KB4	36.	R-QN1	RXP CH
17.	P-Q5	P-QB4!	37.	K-N1	R/B7-KN7 MATE
18.	R-KB3	P-KR3			
19.	N-KR3	Q-R4			
20.	N-B2	N-N4			

(A) Better is 11. ... P-KR3.

(B) White has built up a formidable position by very simple means. It is significant that White has avoided playing P-K5 prematurely when it would have resulted in a weak pawn, but has waited for the moment of maximum effect. Credit the heuristics which encourage full deployment of pieces before undertaking anything adventurous.

(C) A bad mistake due to a programming misunderstanding. The simple QR-N1 leaves White in command. Now Black's game has been freed and White caves in to his opponent's superior ability.

(D) White sidesteps the dangerous N-B6ch in one variation, but his game is positionally hopeless.

(E) 23. ... R-N2 first is better and would win a pawn.

(F) Q-K3 could have given rise to a dangerous counter-attack. But this involved more judgement than the program is capable of.

(G) Finally Black wins a P and for practical purposes ends the game.
(H) This error is due to the quiescence mechanism which only investigates captures but not checks. The program thought the main line was 31. ... RXR, 32. RXR and saw no danger in the move R-B8 (mate).

Game 3: GENIE vs TECH, Second Annual Computer Chess Championship

	GENIE	TECH
1.	P-KU	P-Kl»
2.	N-KB3	N-QB3
3.	B-QN5	N-KB3
U.	0-0	B-QB«*
5.	N-QB3	P-Q3
6.	P-QU	PXP
7.	NXP	B-Q2
8.	N-KB5	0-0
9.	B-KN5	N-KU (A)
10.	N-QRU	BXB
11.	NXB	BXR

and TECH eventually won*

(A) White has achieved a significantly superior position. However this was accomplished by following a pre-stored "book" variation. TECH has stayed In the "book" by making moves selected by Its opening heuristics package. It Is significant to note that this heuristics package Is good enough to generate a "standard" line of play/ and that a program that relies on pre-stored variations of this type frequently (as Is the case here) makes a mistake as soon as the game departs from Its pre-stored knowledge.

Game 4: TECH without positional analysis vs TECH
(Both with 4 ply search plus quiescence)

	WITHOUT	WITH		WITHOUT	WITH
1.	P-KR3	P-K4	17.	B-K2	R-KB1
2.	N-QR3	P-Q4	18.	RXR CH	KXR
3.	N-KB3	N-QB3	19.	P-KR5	P-K5
4.	P-K3	N-KB3	20.	P-KN4	Q-KR5
5.	P-KN3	B-QB4	21.	P-Q3	PXP!
6.	N-QN5	O-O	22.	B-KB1	Q-KB7!
7.	P-QR4	B-KB4	23.	BXP	NXB
8.	B-KN2	N-K5	24.	B-Q2	P-QB3
9.	P-KR4	R-K1	25.	N-QR3	BXP
10.	R-KR2	R-K3	26.	BXB	QXB
11.	R-QN1	N-QN5 (A)	27.	K-QB2	K-K2
12.	B-KB1	NXKBP!	28.	K-QB3	P-Q5 CH
13.	RXN	BXBP	29.	K-QB4	NXP CH!
14.	N-KN5	BXQ	30.	K-QB5	Q-QB6 CH
15.	NXR	PXN	31.	N-QB4	QXN MATE
16.	KXB	Q-K2			

(A) Preparing a rather clever combination which could however have been repelled by 12. P-Q3.

(B) The advantage of the heuristics package which presorts the moves at the top of the tree can be very clearly seen from this example.

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Computer Science Department Carnegie-Mellon University Pittsburgh, Pa.	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

The technology chess program

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Scientific report

5. AUTHOR(S) (First name, middle initial, last name)

James J. Gillogly

6. REPORT DATE 29 November 1971	7a. TOTAL NO. OF PAGES 32	7b. NO. OF REFS 5
8a. CONTRACT OR GRANT NO. F44620-70-C-0107 b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) CMU-CS-71-109	

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Air Force Office of Scientific Research

13. ABSTRACT

A chess program has been developed which plays good chess (for a program) using a very simple structure. It is based on a brute force search of the move tree with no forward pruning, using material as the only terminal evaluation function, and using a limited positional analysis at the top level for a tie-break between moves which are materially equal. Because of the transparent structure, this program is proposed as a technological benchmark for chess programs which will continue to improve as computer technology increases.

Key Words: Chess, Chess Programs, Game-Playing Programs, Artificial Intelligence

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT

DD FORM 1 NOV 65 1473 (BACK)
(PAGE 2)

Security Classification