A FIVE-YEAR PLAN FOR AUTOMATIC CHESS

I. J. GOOD*

TRINITY COLLEGE, OXFORD, and ATLAS COMPUTER LABORATORY, CHILTON

JUSTIFICATION OF CHESS PROGRAMS

Young animals play games in order to prepare themselves for the business of serious living, without getting hurt in the training period. Game-playing on computers serves a similar function. It can teach us something about the structure of thought processes and the theory of struggle and has the advantage over economic modelling that the rules and objectives are clear-cut. If the machine wins tournaments it must be a good player.

The complexity and originality of a master chess player is perhaps greater than that of a professional economist. The chess player continually pits his wits against other players and the precision of the rules makes feasible a depth of thinking comparable to that in mathematics.

No program has yet been written that plays chess of even good amateur standard. A really good chess program would be a breakthrough in work on machine intelligence, and would be a great encouragement to workers in other parts of this field and to those who sponsor such work.

In criticism of the writing of a chess program, Macdonald (1950) quoted a remark to the effect that a machine for smoking tobacco could be built, but would serve no useful purpose. The irony is that smoking machines have since been built in order to help research on the medical effects of smoking. This does not prove that a chess program should be written, but suggests that the arguments against it might be shallow. Many branches of science, and of pure and applied mathematics, have started with a study of apparently frivolous things such as puzzles and games.

* Now at Virginia Polytechnic Institute, Blackburg.

It is pertinent to ask in what way a good chess program would take us beyond the draughts program of A. L. Samuel (see Appendix B). The answer is related to the much greater complication of chess, the much larger number of variations and possible positions. In fact, the number of possible chess positions is about the cube or fourth power of the number of possible draughts positions (see Appendix E). Samuel was able to make considerable use of the storage of thousands of positions that had occurred in the previous experience of the machine, and this led to a very useful increase in the depth of analysis of individual positions. The value of this device depends on the probability that, at any moment in the analysis, we run into a position that has already been analysed and stored. If the expected number of previously analysed and stored positions on an analysis tree exceeds unity, then the expected effective number of positions on the tree is infinite! This remark is of course based on an over-simplified model, as is clear since the total number of possible chess or draughts positions is finite. The remark is also somewhat misleading, since an 'infinite' tree might be deep at the wrong places; but it does give a little insight into the effect of storing positions. (See also (a) on p. 92 and Appendix H, p. 114.) Whereas it is useful to store thousands of positions in draughts, in chess it would be necessary to store millions or billions of positions in order to gain a comparable advantage. Presumably the main advantage of storing positions in draughts accrues from the opening and early middle-game; and in chess, by storing millions of positions, the machine could gain an opening advantage. Apart from this, I have believed for a very long time that a good chess program would need to make use of essentially the same methods as those used by men. The difficulty, the interest, and the challenge, is to formalise exactly what it is that men do when they play chess.

The more the program is based on the methods used by humans the more light it will shed on the nature of thought processes. But for the sake of a clearcut objective I should like to write a program that wins games.

When a man plays chess, he does often recognise situations that have occurred before, but these situations are seldom complete positions, except in the opening and very late end-game; rather they are features of the position, or patterns embedded in it. Thus chess provides an example of pattern recognition, whereas draughts does not to much extent. Nevertheless, several of the techniques used by Samuel should be incorporated.

Chess also provides a better example than draughts of the use of associative memory, since any given position is associated with many situations that have occurred in the chess player's experience. These associated situations suggest strategic or tactical ideas to the player, the strength of the suggestions being dependent on the strength of the associations.

An example of a pattern is when White's king has three unmoved pawns in front of it in the late middle-game or early end-game. This pattern is recognised by every experienced chess player as potentially dangerous, in that there is the possibility of a mate by a rook or queen on the back line,

even in conjunction with a sacrifice. One analysis of the position will then be performed with this theme in mind. This applies more generally to the goals and subgoals that occur to the chess player. Thus there should be 'specifically-directed' as well as 'routinely-directed' analysis.

Another important aspect of chess thinking, also required in most other problem-solving, is what de Groot (1946, 1965) calls 'progressive deepening' of an analysis. Typically an analysis of a position by a human player does not simply follow a tree formation, but contains cycles in which a piece of analysis is retraced and improved. (Compare the not purely hierarchical structure of definitions and perception discussed by Good 1962a, p. 124, and 1965a, pp. 42 and 75.) Also a player continually formulates subgoals and subsubgoals and continually modifies them. In these respects again chess provides a better example than draughts.

As every philosopher and programmer knows, it is not easy to formalise human thought. In fact, when we say that a man has made a judgement, we imply that we do not know in detail how he did it, and this is so even if that man is ourselves (Good 1959). This aspect of judgements is not yet mentioned in dictionaries. The work on automatic chess therefore started when men, such as the musician-chess player Philidor (1749), formulated general principles of play. Many of these principles have become embodied in chess slogans or clichés that need to be taken into account by the programmer (see Section 4.2).

Many of the principles of chess are expressed linguistically rather than numerically. This is much more typical of chess, and of many intellectual activities in ordinary life, than in draughts. Thus chess provides a good field, with a limited vocabulary, for those who are working on language handling by computers. (An example that did occur in draughts, or should have occurred, is mentioned in Appendix B, and curiously enough was overlooked by the strong draughts player who was pitted against a machine. He could have drawn the game, and the method can be readily expressed in words, but only artificially in numerical terms.)

Chess programming can also be justified as sport. £500,000,000 are spent on football pools in the United Kingdom in five years, and each pound must correspond to several hours of study. Further, hundreds of millions of man-hours are expended in watching the game. The expenditure on chess programming will be microscopic in comparison.

The nature of complex games

There is one principle of chess that is so old that the early theoreticians probably did not bother to mention it. It is certainly as old as the game of Go, at least 3000 years, and is typical of all complex games, although stochastic games have additional features. It is the habit of thinking 'if I go there, then he might go there or there, and, in the first case, I might go there, etc.'. In other words, analysis often follows the branches of a tree, with evaluations at the endpoints, followed by back-tracking or iterative minimaxing (see

Appendix A). This process corresponds closely to the formulation of games in 'extensive form' (see von Neumann & Morgenstern 1947, or Luce & Raiffa 1957, Chapter 3). The notion of a game tree applies of course much more generally than to chess. With regard to the definition of the endpoints, it is sometimes said that they should be quiescent positions, but, provided that the outcome of the game is clear enough, an endpoint can also rationally be a turbulent position. For this reason, it is convenient to introduce a notion called 'agitation', which is a modification of 'turbulence' taking into account the probability that the result is clear-cut. (See Appendix H for a discussion of the terminology and its relationship to decision making in general.)

Let us suppose that we can measure, for any position π , both its *turbulence* and also the *superficial probabilities* that the player can win, draw, or lose, p_{W} , p_{D} , and p_{L} . (By a 'superficial probability' I mean a probability based on some evaluation function, without any forward analysis. Perhaps 'surface probability' would be a better term.) Then the decision of whether to regard that position, π , as an endpoint, when analysing a position, π_0 , depends on the following considerations:

(a) The depth (i.e., the number of moves ahead) of the position π from the current position, π_0 . More precisely, the decision depends on the probability that we shall reach π from π_0 . The smaller this probability, $P(\pi|\pi_0)$, or, less accurately, the greater the depth, the more rational it is to treat π as an endpoint of the analysis tree for π_0 . The 'probabilistic depth' of π from π_0 could reasonably be defined as proportional to $-\log P(\pi|\pi_0)$, and the expected amount of information in an analysis, or its effective depth could be defined as $-\sum_{\pi} P(\pi|\pi_0) \log P(\pi|\pi_0)$ summed over all the endpoints of the tree. This is an incomplete entropy, since, usually, $\sum_{\pi} P(\pi|\pi_0) < 1$. This definition suggests itself, but I think the analysis given in Appendix H will turn out to be more useful. The value of storing previously analysed positions could perhaps be estimated in terms of 'effective depth'.

(b) The lower the turbulence of π , the more prepared we should be to treat it as an endpoint, other things being equal.

(c) The more obviously the outcome of the game is decided at π , the more prepared we should be to treat π as an endpoint, other things being equal.

(d) The larger the analysis tree as a whole, as far as we can judge, the more prepared we should be to treat π as an endpoint, other things being equal.

(e) The less time we have left on our clock, the more we should be prepared to treat π as an endpoint, other things being equal.

(f) The less time our opponent has left on his clock, the more prepared we should be to treat π as an endpoint, other things being equal. (This advice is double-edged!)

In other words, we should define a certain function of six variables, monotonic in each separately, and should treat π as an endpoint if this function exceeds some threshold.

Notice the following philosophical point. The superficial probabilities are not strict logical probabilities, since they are based on a suppression of a logical argument, namely a forward analysis. (Compare the discussion of the probability of mathematical propositions in Good 1950a, p. 49.) Since we are forced to make use of superficial probabilities, there is bound to be an element of luck in chess. (There has even been a competition on luck in chess: see Golombek 1966.) From the point of view of strict logic, chess is a game of pure skill, but it is a physical impossibility to avoid some degree of chance. Just as in mathematics we must make use of probabilities based on incomplete analysis. When trying to prove a mathematical theorem, it is necessary to formulate subgoals, and to estimate superficial probabilities for them, in order to find an appropriate strategy. Theorem-proving resembles chess playing in that we have an objective and an analysis tree, or graph, but differs in that a superficial expected pay-off replaces the iterated minimax. minimax idea can come in if we are trying to prove a theorem and we imagine that we have an opponent who wishes to disprove it. The value of our game is 1 if the theorem is true and -1 if it is false. In the proof trees described in the paper by Dr D. C. Cooper the 'and's correspond to moves of the opponent, since we must allow for both branches, whereas the 'or's correspond to our own moves. The minimax (strictly maximin) value of the tree tells us whether the theorem is true, and, if we allow for superficial probabilities at the endpoints of the tree, the minimax value is the superficial probability of the theorem.

The programming of complex games will also exemplify an aspect of practical decisions, since here too it is typical that we are forced to make use of superficial probabilities.

Closely allied to the problem of selecting endpoints on an analysis tree, is the selection of which moves to discard without analysis. This problem is discussed in Appendix H.

SOME POSSIBLE KINDS OF CHESS PROGRAMS

2.1. Chess is a game of perfect information, and is a finite game if the 50-move drawing rule is mandatory (see Appendix D). Hence, from the point of view of the Borel-von Neumann theory of games, chess is 'trivial' (see Appendix A). This shows that, for practical chess, the Borel-von Neumann theory has trivial relevance. But it would be a good exercise to program a computer to do an exhaustive analysis of any position given to it. Such a program could be used for solving two-movers, a program for which has already been written at M.I.T., but it would have little applicability to normal chess. It could be largely written independently of the rules of chess, and could invoke these rules as a sub-routine. It would then have applications to other games and decision problems. If this program is written it should have the option of being stopped at any assigned depth.

A program for solving ordinary 'two-movers' would be useful when dealing with a large number of problems, for a tournament or book, or for a writer of chess columns in newspapers. Professor R. C. O. Matthews points out (private communication) that it could be used in the composition of problems.

2.2. If our object is to learn as much as possible about learning, we could try to program a machine to learn the game without even telling it the rules. This is how Capablanca learned chess: by watching his father and uncle playing. Incidentally he won the first game he played, at the age of 6, against his uncle. McCulloch (1965, p. 199) calls a machine of this kind an 'ethical' machine. It would need to be able to formulate a very wide class of hypotheses.

2.3. Next, we could give the rules to the machine and let it learn the principles for itself. Here again the machine would need to be able to formulate a wide class of hypotheses, but not as wide as the 'ethical' machine.

2.4. Next, we could give the machine a limited class of principles and allow it to work out its own evaluation functions in terms of these principles, in the light of its experience. The experience could either be the playing of games or, more efficiently, we could tell the machine which moves in various positions were good or bad, and how good or bad. The simplest form of evaluation function would be linear, with coefficients that were allowed to be positive, negative or zero. The optimisation of these coefficients constitutes a primitive form of learning, but I think most people would be reluctant to call it 'concept formation'. If however non-linear polynomials are permitted. or equivalently if logical combinations of the principles can be formed and incorporated into the linear function, then concept formation becomes possible. For example, a machine might have discovered that two bishops are usually better than bishop and knight, it if had been programmed to examine quadratic evaluation functions (Good 1959). After this had been established, it might have gone on to modify the concept by saying that it was true in open positions, but not in closed positions. It might have discovered the whole of this principle in one step if it had been allowed to use cubic evaluation functions, but this would be a more expensive way of making the discovery. It seems reasonable to say that the use of quadratic evaluation functions provides the first step in automatic creativity or concept formation.

Humans usually think this way. First they give relative weights to features, then they look for interactions between pairs of these features. When they find a pair of features that need to be taken together, they define this as a new feature. Thus high-order interactions are often discovered piecemeal, but of course this is not always possible.

The discovery of more intricate concepts will require the generation of a large number of propositions, and the order in which they should be generated provides a most interesting search problem.

2.5. We could try to write the program with man-machine synergy in mind from the start (regarding the terminology, see Appendix K). We would aim to teach the machine how to play chess, and also to learn something about chess from the machine. Our long-term aim would be to cooperate with the machine in games against grand-masters, against other machines, and against other synergistic combines. A simple and effective example of man-machine

94

synergy is provided by R. C. O. Matthews' suggestion mentioned in section 2.1.

2.6. We could go all out to make the machine a good chess player, by putting as much as we can into the low-level program, and without relying on more machine learning than is implied in the optimisation of linear forms.

2.7. Same as 2.6, but with the machine capable of forming new concepts, that is, allowed at least to use non-linear evaluation functions.

Of these various aims, which are not mutually exclusive, the one under heading 2.2 is the most exciting, but 2.6 would be easier. I think a reasonable approach is to aim at 2.6 at first, in the hope of getting a good chess program, but always holding the other aims in mind.

A BRIEF HISTORY OF CHESS PROGRAMS

As mentioned earlier, the formalising of principles for playing chess began with the early theoreticians. But they wrote exclusively for people, just as grammarians wrote for people and not for machines until the advent of computational linguistics. Emanuel Lasker (1932, p. 339) said 'It is easy to mould the theory of Steinitz into mathematical symbols by inventing a kind of chess, the rules and regulations of which are themselves expressed by mathematical symbols. The Japanese game of Go is very nearly what I mean. In such a game the question, whether thorough analysis would confirm the theory of Steinitz or not, presumably could be quickly solved because the power of modern mathematics is exceedingly great.' What Lasker had in mind was clearly not a mathematical method of playing chess, but a mathematical demonstration of the validity of strategic principles for wider applications. Incidentally, I believe that a good Go-playing program would be much more difficult to write than a good chess program, because Go, not to be confused with Go Moku, depends so much on judgement. Perhaps Lasker was misled by the charming simplicity of the rules of Go (see Good 1965b).

In about 1932, one of my schoolboy chess associates named G. T. Hammond pointed out to me, without claiming originality, that there is a simple mechanical method of playing that is not as bad as one would expect, at any rate in the opening. The method is to play the move that maximises one's *mobility*, where mobility is defined as the number of possible legal moves. I cannot remember whether he allowed for the mobility of the opponent. We realised that this rule should not be used if it involved putting a piece *en prise*, and that the rule was not much good in a combinative position. But this observation about mobility is I think the most surprising single fact, so far, about automatic chess. It might date back much further than 1932, since the theoretical valuation of the pieces, of 1876, was based on the average numbers of squares controlled (see Appendix F). Compare also a remark attributed to Emanuel Lasker by Edward Lasker (1951, p. 56), '... the player who gains control of more than 32 squares has the better winning chance'.

Good (1939) remarked that it might be possible to translate into a mathe-

matical technique the knack that masters have of picking out only those lines that matter from the millions of possible ones. This is of course the central problem in writing a good chess program. (When that note was written, what is now called 'programming' would have been called 'mathematics'.) Another point in that little article was that some statistical reasoning could shed light on the nature of chess. For example, White's advantage in moves is $1-1+1-\ldots$, which is summable (C, 1) to $\frac{1}{2}$. In other words, White is, on the average, half a move ahead. Since $2\frac{1}{2}$ moves are worth a pawn, it follows that White's advantage is about a fifth of a pawn. Now, by statistical sampling we could quantify what I called the law of multiplication of advantage, and, from this, we might be able to shed light on the question of whether White has a theoretically forced win.

In 1940, when I met and worked with Turing, we had several discussions on how chess playing could be formalised. We also later had discussions with Shaun Wylie and Donald Michie. Essentially similar ideas were published by Shannon (1950), who had been thinking independently about chess programming. Turing, Michie and Wylie, in 1947-48, designed some onemove analysers, a discussion of which was published by Maynard Smith and Michie (1961) and Michie (1966).

In about 1941, I had discussions with J. Gillis on the possibility of founding a theory of imperfect logic, in which propositions would be exponentially forgotten if not used for some time, and also could be reinforced (the aim was to formalise human logic). A similar idea has been used very effectively by Samuel in his draughts-playing program.

Perhaps the most intensive work ever done on the formalising of chess thinking was that of de Groot. He studied the psychology of chess playing for over twenty years (see de Groot 1946, 1965). I have not yet been able to judge how much of the later literature is explicit in de Groot's writings. His basic experimental method was to record the spoken moment-to-moment thoughts of chess masters.

Shannon (1950) made the following points:

(a) Different routines should be used for opening, middle-game and ending.

(b) Forceful variations should be analysed out as far as possible and evaluated only at points of some stability.

(c) Select the variations to be explored by some process so that the machine does not waste its time on totally pointless variations. (This point was covered by Good 1939.)

(d) The threshold for deciding whether a position should be regarded as stable should depend on the number of moves advanced from the current position.

(e) A statistical element can be introduced in the selection of moves so that an opponent who has won once cannot win again by exactly the same sequence of moves.

(f) A few hundred openings could be stored and played by rote.

(g) A learning program could be based on the use of a higher-level program

that changes the terms and coefficients in the evaluation function, according to the results of games.

Slater (1950) argued that 'it does seem possible that a chess computer which was programmed, beyond immediate tactical tasks, to maximise the mobility difference between itself and its opponent over a series of moves, might play a strategically tolerable game of chess'. His thesis was based in part on an examination of master games. (I do not know whether G. T. Hammond's remark was so based.) He also gave some statistics relevant to the law of multiplication of advantage.

Good (1950b) pointed out that 'the analysis of forceful variations in any position would be like a stochastic branching process, and the endpoints of the corresponding tree would be quiescent positions needing strategic evaluation (but see Appendix H). The total time taken to cope with such a tree would be roughly proportional to the number of individuals in this tree, and it is known that the probability distribution of this number is extremely skew (see Good 1949).... There would therefore be a danger of the machine running into time trouble. It would then modify its definition of the forcefulness threshold of a move.'

Good (1959, based on a lecture of January 1958) pointed out that:

(a) The training of a machine should be move by move, and not merely by the results of games.

(b) In quiescent positions it should be possible to find some rule for estimating the probability of a win, draw or loss, and this is analogous to the estimation of the probability of a mathematical theorem. 'It may be sufficient to assume that the log-odds are linear in all the types of advantage that are worth listing... The probabilities of winning, drawing and losing can perhaps be expressed in the form $(p^2, 2pq, q^2)$, and the above log-odds mean $\log(p^2/q^2)$. Judging by some statistics of E. T. O. Slater, p^2/q^2 is approximately 1.4 for the initial position in master chess. I believe that White's advantage is worth half a move or a fifth of a centre pawn, so a pawn is worth a factor of $1.4^5 = 5$, and a queen $5^{9i} = 5,000,000$.'

(c) More convincing than the determination of the coefficients of a linear form, as an example of judgement, would be the determination of a functional form. The example of two bishops was mentioned.

(d) Very few human chess players discover new strategic principles for themselves. If we ask this of a machine we are asking for a level of creativity well above the average for humans.

(e) One advantage of an evaluation function expressed in terms of probabilities is that it could be worked in with estimates of the probabilities of the opponent's making various moves in any given position.

In 1956 a program was written at Los Alamos (Kister *et al.* 1957), capable of occasionally beating a beginner. The game was on a 6×6 board, and the analysis was exhaustive to two moves on each side. The evaluation was the simplest possible: a sum of material and mobility measures.

Bernstein et al. (1958a, b) wrote a program for the full 8×8 board, also

analysing two moves ahead, but with considerable move selection. As Newell, Shaw and Simon (1958) point out, this selectivity leads to a great increase in the complexity of the program, and so to a great slowing down *per move within* an analysis but that, in a deep analysis, this slowing down is far more than compensated because the number of variations to depth d goes up exponentially with d. If the average of the number of moves to be tried per position is *appreciably* decreased then the size of a deep tree is *enormously* decreased.

Newell, Shaw and Simon (1963, p. 47) mention en passant a hand simulation by F. Mosteller, and a Russian program for the BESM, but had little information about these two programs. Their own program avoids the use of a single evaluation on the grounds that humans do not seem to use one. They are especially concerned with the simulation of human thought, and they believe, as I do, that for a good chess program it is necessary to be so concerned. A number of goals are set up, and there is a move generator for each goal. Thus the evaluation function can be thought of as a vector. The goals that are listed are king safety, material balance, centre control, development, king-side attack, and promotion. Centre control and material balance are discussed in detail. Although the evaluation functions are not scalars, it is necessary to decide on an ordering for them in order that iterative minimaxing should be possible. One ordering suggested is lexicographic, the material balance being taken as the first component of the evaluation vector. It seems to me that any strict ordering of evaluation vectors effectively converts them into scalars. In lexicographic ordering, the components of a vector correspond to the digits of a scalar expressed relative to a large radix. The scheme is thus economical for the same reason that variable-length arithmetic sometimes is.

A recent reference is Newell and Simon (1965).

OUTLINE OF A FIVE-YEAR PLAN

The suggested plan can be split up into a number of parts, which I have listed here very roughly in the order in which they would be carried out. But this plan would be subject to revision from time to time. 'Progressive deepening' applies here just as much as to the analysis of a chess position! Note that the plan has not yet reached the software or hardware; it is in the form of so-called 'supporting underware'.

Five years seems a reasonable estimate for the project for a team of about three people working full-time. In the following breakdown of the work, I have not yet tried to allot these fifteen man-years.

4.1 Program a computer to play legal chess. This has of course been done several times before. It is not as easy as it sounds. A 'position' must be defined to include the information of whose move it is; whether the kings have moved, and, if not, whether the rooks have; whether $P \times P$ en passant is legal; how many times the position has occurred before with the same player to move (where here 'position' has its naive interpretation, and presumably no distinction is made between, for example, White's KR and QR); and how

many moves have been played since the last capture or pawn move. A move that gives check must be so described. A distinction must be made between the making of a move and its mere consideration. The legal-chess program should be efficient, since the speed of the *whole* chess program will be proportional to the speed of this one.

The lists that are worth keeping are those that are cheap to up-date. These should include a list of all squares occupied by White and by Black. Then, when we are considering White's moves, we do not need to run right through the 64 squares of the board to find the white men. (For the purpose of the whole program, for playing good chess, not merely legal, we should have lists of squares dominated by the two players.) Whether we are in check, and whether it is a double check, should be recorded from the opponent's previous move. If, for example, we are in double check, we know that we must move our king, in virtue of a theorem that happens to be true for chess although it would not necessarily be true if the rules were slightly different. If we are not in double check, then lists of our pinned pieces should be made before looking for legal moves, together with the rays along which they are pinned, and also whether each pin is relative or absolute, i.e., whether the ray is one along which the piece can or cannot move. There should be a transient list of our own pieces, from which we cross off each piece after we have listed all the legal moves we have made with it from the given position.

4.2. Collect chess principles from good books on chess, for example, Lasker (1932), Fine (1941), Horowitz and Mott-Smith (1960), Euwe and Kramer (1964-65), Suetin (1965), and de Groot (1965); by introspection and memory; and by dictating your chess thinking to a secretary or dictating machine. Each principle constitutes a potential goal. Here is a preliminary list.

Analyse 'forced' variations either to quiescence or to a point where the result of the game is obvious. A forcing move is one that gives little option to the opponent on his next move, and includes all checks. There are degrees of forcefulness, and the threshold on the forcefulness will vary according to the depth of the analysis from the current position. More precisely it should allow for the discussion of Appendix H concerning the meaning of an 'endpoint' in an analysis tree.

A special case of the above principle is the following deep thought: if you can mate on the move, do so.* Also, never miss a check, it might be mate, a motto attributed in jest to the Hampstead Chess Club by J. F. O'Donovan. More generally, consider those moves that restrict the opponent's choice on the next move, even if this involves a sacrifice; but don't stalemate him except on purpose.

Material values: one conventional set is P=1, N=B=3, R=5, Q=9, but

* This could be extended: if you can mate in two moves, do so. For a complete analysis as far as your second move would be cheap on a large computer. In some positions an exhaustive analysis to n moves deep, for n > 2, could be carried out by calling the subprogram mentioned in Section 2.1.

B+B=7. In particular, 'sacrifice' your queen if you get value for it. The advancing of a passed pawn can be regarded as the obtaining of extra material. For on the eighth rank a pawn is worth a queen. My own rule is that a passed pawn that cannot clearly be stopped is usually worth about five pawns on the seventh rank, three on the sixth, and one and a half on the fifth. A pair of double pawns=1½. 'The solution to every end-game is "pawn on" '-Doog. Swap pieces, but not pawns, when ahead in material, but don't let the position get completely blocked. See also Appendix F for values of pieces and squares.

In the opening, pawns can be arranged in decreasing order of value, according as they are Q or K pawns (central pawns), B pawns, N pawns, or R pawns. Lasker (1932, p. 107) gives the ratios 8:6:5:2, but I should like to see a proper statistical survey of games to bear this out. Backward or isolated pawns are usually weak. If your opponent has them, blockade them with pieces. Two pawns, side by side, are strong because they cannot be blockaded. Try to fix a group of your opponent's pawns with a smaller number of your own. Avoid putting pawns on the same coloured squares as your lone bishop, especially if these pawns can be blockaded.

Increase your mobility, as compared with your opponent's. As discussed earlier, this includes more than is immediately apparent. To some degree it covers, for example, the advice to control space, and especially the centre, to get the pieces developed, and to move rooks to open and half-open files.

Attack the bases of your opponent's pawn chains, and defend the bases of your own.

Don't open up the position when you are behind in development.

Prevent your opponent from doing what you would do in his position. Decrease his mobility by advancing pawns. Do as you wouldn't be done by.

Exert pressure on the squares around the opponent's king.

The best strategy against a pawn storm on the wing is usually a thrust in the centre. If the centre is blocked and the kings are castled on oppositesides, then a pawn storm against the opponent's king is indicated. But if the pawns protecting the king are side by side, it is advisable to force one of them to advance before making the pawn storm.

When an attack fails, the side that was attacking is liable to be badly placed for the defence (as in Go), so don't attack without adequate reason.

Strengthen the weakest link in your position when defending, and attack the weakest link in your opponent's position when attacking. Move piece in worst plight. Exchange aggressive pieces when defending. Look for counterattacks.

Preserve options, for example, develop at least one knight before both bishops in the opening, since the knights usually have fewer squares to go to.

Get pieces to aggressive positions if they can be held there; for example, knights on the fifth and sixth ranks on central or bishops' files, and rooks on the seventh rank for winning pawns and restricting the opponent's king.

Make your pieces co-operate. Co-operation means the achieving of sub-

goals by groups of pieces. For example, have a pawn protecting your advanced knight; connect your rooks by castling; double your rooks on open file or seventh rank. Avoid blocking your pawns with pieces.

Try to make double threats, as in many games.

Make each piece as efficient as possible without over-burdening it. The most important work should be done best if the costs are equal. For example, overprotect the centre, so that the pieces doing this work will have flexibility. (This is again a matter of mobility, but not according to the formal definition in terms of number of squares attacked.) Generally, flexibility is important. (See also the last sentence of this sub-section.)

In quiet positions, aim to accumulate small advantages. This should also be done in turbulent positions, but it is then more difficult to judge. (It needs a master's skill to carry out a complicated combination in order to win half a pawn!) If your small advantages are otherwise transient try to convert them into more permanent ones. Look harder for combinations, that is, lower the threshold of 'forcefulness', when your small transient advantages add up to as much as say 2/3 of a pawn, and also if your opponent has weaknesses such as cramped pieces, unprotected pieces, exposed king or queen, weak squares in front of king, queen and king in a line. The greater your nonmaterial advantage, the lower the threshold of 'forcefulness' should be.

Standard won classes of end-games should be in store, since they provide endpoints on the analysis trees, and serve a similar function to the stored positions in Samuel's draughts program. A classification of mating positions and motifs serves a similar function (see Appendix G).

The following principles apply especially to the end-game.

An extra pawn is usually enough to win, if there are pawns on both the Q and K side of the board.

If each side has one bishop and these stand on squares of opposite colours, then an advantage of one pawn is usually not enough for a win.

A B is better than a N when there are pawns on both the Q and K side.

K and Q beat K; K and R beat K; K and two B's beat K; K, B and N beat K; K and two N's do not beat K; K and B beat K and 32 B's all on white squares!

If you are two pawns up (in the end-game) you can force a win by advancing them, since your opponent will have to give up a piece for them. This is a special case of the Law of Multiplication of Advantage.

K and P beat K in situations that could be completely formalised without great difficulty. For the present, note that it is useful to get the K in front of the P, and to get the 'opposition' (as in draughts).

K and Q usually beat K and R. The method, which is easy to express in language, but not to formalise, is to bring the K and Q close to the opponent's K and R, force them to the edge of the board, and, by threatening to win the R by pinning it, to force it away from the K. Then it can be forked or mate given.

A rook's pawn is stronger against a knight than other pawns are.

K, B, and RP beat lone K if the corner in front of the pawn is one the B can go to.

Put rooks behind your own passed pawns.

KUFTEG (king up for the end-game). For the king is worth four pawns in the end-game and should be used. If you are not yet sure where to use it, put it in the centre. It will then have less distance to go than it might have had; in other words the king, like the knight, has more flexibility when it is in the centre.*

4.3. Review the literature of chess programs to see if it contains any principles not already in the literature of chess.

4.4. Classify the various principles, in order that later formalisation should be more efficiently performed, and in order to find gaps.

4.5. See if the literature of learning theory, concept formation, pattern recognition, perception, and associative memory suggests new principles. If so, revise step 4.4. Don't devote five years to this step!

4.6. Try to express the various principles unambiguously and, where possible, quantitatively.

4.7. Devise a special-purpose chess language for expressing the principles as unambiguously as possible, but with not too large a vocabulary. (See Newell & Simon 1965, p. 62, for a preliminary list of chess vocabulary.) Complete lack of ambiguity will not be essential if the application is manmachine communication, since the machine might be able to make syntactic transformations without full grasp of the semantics. Ambiguous statements can be stimulating and concise, for people.

4.8. See if the terms of this language can be expressed in Algol or similar language.

4.9. Make a study of chess planning, to see whether it requires a language different from the expression of the chess principles. For example, one might say, 'Since the position is closed, I have time to get my knight round to Q5'. This will involve us in the study of the minimal syntax required for the convenient expression of chess principles and planning.

As an example of planning, consider the position in Fig. 1, from Lasker (1932, p. 24):

K.k.... P..... N.....

FIG. 1. White to move. An example in which an efficient analysis is linguistic. Notation: lower case for Black.

* It is a universal principle of rational behaviour that flexibility has to be sacrificed when a decision is made, but a decision to *acquire* flexibility can be made. There is no contradiction here, since 'flexibility' can be achieved at various levels of a hierarchy.

A human might reason in the following manner. Could a machine do this? 'My opponent has only a king and I have a knight and pawn, so I can win only if I can promote the pawn (goal). In order to advance the pawn I must move my king to N7 or N8 (subgoal). This is prevented at present by the black king. He can continue to prevent it (his goal is to do so) by moving his king to B2 and back to B1 and so on (his plan). I must move my knight into such a position as to control his B2 when his king is at B1, or to control his B1 when he is at B2 (subsubgoal). Consult list of theorems about knights. Each time they move, the colour of their square changes. Ah, but each time the black king moves the colour of its square changes (inspiration). Therefore I can never succeed in my subsubgoal, and cannot win, if he continues with his plan.'

If we could put plans in order of merit, then plans, counterplans, countercounterplans, etc., would form a *tree of plans*, and iterative minimaxing would apply to it.

4.10. Estimate the cost of having built-in facilities for making chess programming easier, on a general-purpose computer. Consider whether some of these facilities could be generalised so as to be useful for other kinds of problem-solving. For a review of more general problem-solving, see Newell and Simon (1961). Remember that the problems of hardware and software are inseparable. General problem-solving programs should not be divorced from general problem-solving machines.

4.11. Plan the effort so that it can be broken up into self-contained pieces, for clearer understanding and for parcelling out.

4.11.5. Work out how to organise a program so that positions that occur on an analysis tree, together with their analyses, are held in store for several moves in the course of a game, until the chance that they will occur in later analyses in the game becomes negligible (see also Section 4.23 below).

4.12. Write separate programs for combinations, for the openings (see Appendix J), for the middle-game, and for the end-game, just as separate books are written on these aspects of the game. What other aspects merit separate treatment?

4.13. Write out routines for winning several special end-games, without analysis trees, that is, by using procedures already known to chess players.

4.14 (continuing 4.8). Write a compiler in Algol, or in some other programming language, or in some modified form of an existing programming language, for the programs written in the chess language.

4.15. Assemble theoretical and experimental arguments for the evaluation of the relative values of the pieces, squares, and other features of the positions (see Appendix F). Can these be related to the changes in the log-odds of winning as against losing, or some such probabilistic measure? (see the discussion of Good 1959, in Section 3 above). Is there a general measure for the turbulence of a position? (see Appendix H). Chess programming and mathematical theory about chess should go hand in hand, but the programming will involve more work. Given the number of times that each player

controls each square, in a quiescent position, what is the player's expected advantage?

4.16. Work out further theory concerning optimal searches in trees. Is it possible to make a tentative evaluation of a chess or draughts *program* by calculating the statistical distribution of size of the analysis tree for given rules concerning the thresholds of the agitation at the endpoints? How should the thresholds for selecting endpoints on the trees vary with depth, with agitation, with the amounts of time left on the two faces of the chess clock? (see Appendix H). How should we estimate the value of storing previously analysed positions?

4.17. How should goals be generated? Each potential goal on a check-list can be considered, and rules worked out for assigning relative importance to them. I think we should aim at estimating the probability that we can achieve each goal, and the utility to us if we achieve it (see Appendix C). For this purpose, I think 'progressive deepening' is necessary (see Section 4.23 below).

4.18. How should the move-generators be organised, i.e., what moves should be ignored?

4.19. How should a variety of evaluation functions be handled, each appropriate to a different goal? Can evaluation functions be conveniently adjusted, during an analysis, instead of recomputing them in detail for each move considered?

4.20. Select about fifty quiescent chess positions, get a chess master to put them in order of merit for the player (that is, the side with the move), and then do the same with a variety of evaluation functions. Try to find an evaluation function, partly by hill-climbing, that puts the positions in nearly the same order as the chess master. If there are any bad discrepancies left over, think again and recheck with the master.

4.21. Correlate evaluation functions with those obtained by analysis plus iterative minimaxing, with the object of automatically improving the coefficients (see Samuel 1959, 1963).

4.22. Find evaluation functions that indicate whether a position cries out for startling moves, that is, moves that would be rejected, on a superficial analysis, by most of the goals. Without such evaluation functions, the machine will not be a genius.

4.23. Formalise the process of 'progressive deepening'. This is especially important for problem-solving in general. Example: the threshold at various depths of analysis should depend on how large the analysis tree is going to be. So we should first do a quick pilot analysis, for estimating the size of the tree, and then a more comprehensive analysis. There should be several stages in this process, since no simple method is likely to work. The difficulty is that a single branch of a tree can be greatly enlarged as a consequence of a slight change in the thresholds. The amount of the enlargement cannot be predicted. Hence pilot analysis might have to be applied to subtrees, as well as to the complete tree.*

* The device of holding in store positions that occurred on the analysis trees earlier in the game (see Section 4.11.5) will be relevant to pilot analyses.

Another application of progressive deepening is that ideas that occur at one stage can be used again at a later stage. For example, a move that was considered at one stage and seemed 'interesting' has a better chance of coming up for consideration at a later stage, whether or not it was sound at the earlier stage.

4.24. The relevance of progressive deepening to the formulation of subgoals should be worked out, or at least worked on.

4.25. Play a few games and make improvements in the program as they suggest themselves.

4.26. 'Conversations' should be held with the computer in the special chess language (man-machine synergy). See what modifications in the chess language are required in order to answer new types of questions about chess. For example, we can ask the machine to count the number of distinct games in which White mates on his third move. (According to an analysis I did in about 1932, the answer is 367.) Another example would be retroactive analysis.

4.27. Try to find a transformation that converts an evaluation function into $p_W - p_L$ (or more generally into $p_W - \lambda p_L$), where p_W and p_L are the probabilities of winning and losing respectively. Try to do this even for turbulent positions (see Appendix H).

4.28. Investigate how the analysis should be modified in order to allow for the strengths, weaknesses, knowledge, habits, and style of a particular opponent.

4.29. Attempt to generalise the entire plan to a wider class of games. Here 4.26 will have been useful practice. Begin with randomised chess (see Appendix J), new and unusual kinds of pieces, and three-dimensional chess (see, for example, Good 1957). Many interesting variations on chess are known, such as Kriegspiel (which is not a game of perfect information), Buying Chess (Landau), Panzerspiel, Shotgun, chess with extra kings, chess without kings, chess on a torus, chess with mined squares, losing chess, and chess with a gradually increasing number of moves of each side. Randomised chess is special, since it is intended to replace ordinary chess, i.e., ultimately to be called 'chess'.

Note that, owing to limitations of human spatial visualisation, the machine might play better in three dimensions, relative to a man, than it would in two dimensions.

APPENDIX A. THE BOREL-VON NEUMANN THEORY OF GAMES, AND ITERATED MINIMAXING

In the 1920s, Borel and von Neumann founded a theory of games, which reached large book size in von Neumann and Morgenstern (1944). For the early history, and kudological remarks, see Borel, Fréchet and von Neumann (1953). In the zero-sum two-person game the two players independently choose two 'plays', *i* and *j*, which can be identified with the choice of a row and a column of a matrix, and the pay-off to the first player is a_{ij} and that to the second player is $-a_{ij}$. Neither player knows the other's play. The matrix (a_{ij}) is called the *pay-off matrix*. Chess is a special case of a zero-sum two-person game if we imagine that, according as the

first player loses, draws or wins, he collects -1, 0, or 1 from the second player. The play, *i*, is then to be interpreted as a set of rules which uniquely determine the player's move in each position, so that a play is not just a chess move, but is a single-valued function of positions taking values that are moves. For the selection of an optimal play, 'all' that is necessary is to work back from the end of the game, and deduce in turn whether each position is a theoretical win, draw, or loss for white. But the number of possible chess positions is about $10^{46\pm3}$ (see Appendix E) so an exhaustive analysis is virtually a physical impossibility. Perhaps the use of probability ideas in chess has the same kind of justification as its use in classical statistical mechanics!

The Borel-von Neumann theory can be applied to chess in a practical way only if the game is regarded as probabilistic, and then only in an exiguous manner. The theory shows, for example (what is obvious), that, if we can evaluate the position at all move-pairs from the current position, and if these evaluations are regarded as expected pay-offs, then we minimise our expected loss by playing the minimax move. This is the best move for us on the assumption that our opponent plays optimally, otherwise our best move is the one that maximises our expected gain after allowing for a probability distribution over all the moves that our opponent might make. When we do not assume that our opponent will play optimally, we are playing either psychological or trappy chess or both. All this can be extended to deeper analysis of a position, and leads, when we assume our opponent is playing well, to mini-maxi-mini-maxi-mini-maxing, etc., conveniently called iterated minimaxing, although the epithet 'iterated' is often omitted in discussions of chess and draughts. For psychological chess, a more appropriate term is (iterated) 'expectimaxing', a term that Michie (1966) applies to games against Nature, or against an opponent who moves randomly. Minimaxing is a special case of expectimaxing from the present point of view.

Since we associate +1 with a win for us, it would be more correct to use the term 'maximining' (and 'maxi-expecting') but 'minimaxing' is standard.

APPENDIX B. SOME COMMENTS ON SAMUEL'S DRAUGHTS (CHECKERS) PROGRAM

Samuel (1959) described a draughts program which is capable of sometimes beating near masters. His objective was to experiment with a learning program and he used an evaluation function whose coefficients could be varied by the program.* He resisted the temptation to store book variations, but he did store positions that occurred often enough in the experience of the machine. Presumably the effect of this 'rote' learning is that the program learns the openings to considerable depth, especially if it were pitted against a fixed player for a larger number of games in succession. (It can forget positions as well as learn them.) This device would be much less useful in chess since it has far more openings and not many of these extend almost to the end-game as they do in draughts.[†]

Samuel's paper was reprinted in 1963, together with the beginning of a game played in 1962 against Robert W. Nealey, who had been Connecticut champion, but was not at his top form. Unfortunately the end of the game was omitted in error. The position at the point where the record ceased is shown in Fig. 2 (i). The remainder of the game has been communicated to me by Dr Arthur Samuel, and is: 12-19, 32-27; 19-24, 27-23; 24-27, 22-18; 27-31, 18-9; 31-22, 9-5; 22-26, 23-19;

* It was not clear that these coefficients had reasonably well converged after several games. This makes one wonder how much of the evaluation function is really relevant to the success of the program.

† It would be interesting to have a typical frequency count (n_0, n_1, \ldots) , where $n_r =$ number of stored positions in the draughts program at depth r, from the initial position.

26-22, 19-16; 22-18, 21-17; 18-23, 17-13; 2-6, 16-11; 7-16, 20-11; 23-19, White (Nealey) resigns.

| | | (| (i) | | | | | | | (| ii) | | | | (iii) |
|-----|----|---|-----|---|---|---|---|---|---|-----|-----|---|---|---|---------------------------------------|
| ••• | х | • | х | ٠ | • | • | • | • | х | • | • | • | • | • | O .xO. |
| ••• | •• | X | • | • | • | • | • | • | • | х | • | X | • | 0 | • • • • • • • • |
| х. | • | ٠ | • | • | • | • | • | • | • | • | • | • | • | • | oX |
| . 0 | • | ٠ | • | X | • | • | • | 0 | • | • | • | • | • | 0 | X |
| ο. | • | • | • | • | • | • | 0 | • | • | • | • | • | • | • | |
| ••• | • | ٠ | • | 0 | • | 0 | • | | • | Х | • | • | • | | |
| • • | • | ٠ | 0 | • | • | • | • | • | • | • | • | • | • | • | • • • • • • • |
| . 0 | ٠ | • | • | ٠ | ٠ | • | • | • | • | • ' | ٠ | • | • | • | • • • • • • • • • • • • • • • • • • • |

FIG. 2. Positions (i) and (ii) occurred in the game Machine v. Nealey, 1962; and position
(iii) would have occurred if Nealey had played differently on his penultimate move. Notation: Black, crosses; White, noughts. Kings are in upper case.

In my opinion, White could have drawn the game if he had played 16-12 near the end in place of 16-11, which is a tactical blunder (see Fig. 2 (ii)). He would then have made a king which he would have been forced to exchange, but he could soon make another one. Black will be able to make a second king, but he will not be able to dig White's king out of the double corner, since White, in Fig. 2 (iii), can move his other king backwards and forwards while Black has his kings in the position shown, or further away. The interesting thing about this analysis, as mentioned in Section 1, is that it is not easy to see how it could be expressed in numerical terms.

In a letter of April 19, 1966, Dr Samuel remarks that, in a return mail match, Nealey won one game and drew five. Also that the machine played four games against each of W. F. Hellman, the world champion, and Derek Oldbury, a British champion, and that the machine lost all eight games. The machine, the IBM 7090, was held to five minutes per move at most, whereas the humans had no time limit. It seems reasonable to describe the machine as a near minor master, especially as the winning players required 70 to 80 moves to achieve winning positions. My guess is that the machine is very sound tactically but lacks judgement.

APPENDIX C. WHAT IS THE 'BEST MOVE'?

'Theoretically' we can do a complete analysis of any given position, π , and arrive at the endpoints of the tree, whose values (win, draw, or loss, for the machine) are known directly from the rules of chess. Then we can backtrack, by iterated minimaxing, to get the values of all the positions that can be reached from π in one 'half-move', i.e., one move by the player whose turn it is. In this way we can in principle determine the moves that give nothing away, and, from an over-logical point of view, we could describe these moves as 'equal best'. Then again, we could instead define the best move (or equal best), if π is theoretically won, as a move that forces checkmate in the minimum number of moves against any play however good. The best move in a lost position would analogously be defined as one that delays mate for as long as possible. If π is a theoretical draw, then in much the same spirit we could define the best move as one that delays for as long as possible an indisputable draw (assuming the 50-move drawing rule to be mandatory). For since, at the moment, we are assuming the machine to be a perfect player, we should like the opponent to have as many opportunities as possible for making a mistake. Or it might be a little more logical to maximise, not the number of moves to an indisputable draw, but the product of the number of losing choices that the opponent would have at each of his moves. For this is the total number of ways he has of

going wrong. It is a little more realistic simply to maximise the probability that he will go wrong, and this brings us close to a practical definition of 'a best move'.

Suppose that, as a result of each of the moves the machine can make in position π , we can estimate the (superficial) probabilities p_w , p_D , p_L , that we (the machine) will win, draw or lose. For a given state of the tournament, prize-list, etc., these possibilities will have various utilities, u_w , u_D , u_L , for us. The expected utility of the move will therefore be of the form $p_w u_w + p_D u_D + p_L u_L$. But since $p_w + p_D + p_L = 1$, this expected utility is of the form $\alpha p_w - \beta p_L + \gamma$. Since we are concerned only in finding the best move, we can drop the constant γ , and also divide out by α . Hence the expected utilities, up to an irrelevant linear transformation, are of the form $p_w - \lambda p_L$. A similar argument can be applied by the opponent, so that, if for the moment we regard the probabilities as objective, we can infer that $\lambda = 1$ if the game is 'zero-sum'. So it is fairly reasonable to take the expected utility as $p_w - p_L$; but $p_w - \lambda p_L$ is more reliable.

If our evaluation functions enable us to infer the value of $p_w - \lambda p_L$, then we could 'in principle' backtrack and find the best moves in the sense of expected utility. If we were dealing with a game, like Hex, in which there are no draws, then the best move in the sense of the principle of rationality would also be the best in the sense of maximising the probability of winning.

There is a slight over-simplification in the above discussion since, in many tournaments, there is more utility in winning quickly than slowly. It preserves your energies for the next game. I have ignored this effect in this paper.

APPENDIX D. THE NUMBER OF POSSIBLE GAMES OF CHESS

If fifty moves are played on each side without any captures or pawn moves, then, in some sets of rules, the game is drawn, whereas, in other sets, the draw must be claimed. If the rule is taken as mandatory, then no game can last more than 6000 moves on each side. Also, in any position, even if all the pawns have been promoted to queenhood, the number of possible moves cannot exceed 321. Therefore the number of possible games is less than $321^{12} 000 < 10^{30} 000$. (This inequality was given by Good 1939. With a little more work, the upper bound can be lowered somewhat.)

If we restrict our attention to reasonable games, or to master games, we get a much lower estimate. The method of estimation is similar to one used in certain Monte Carlo estimation problems having no reference to games (see Good 1954). If a_n is the expected number of moves, at a given standard of play, at the *n*th ordinal position, counting the initial position as the first, then the total expected number of games of not more than 2m moves on each side, for *m* less than say 200, whose moves are all of the approved standard, is about

$$\sum_{n=1}^{2m} a_1 a_2 \ldots a_n (1-\epsilon_1)(1-\epsilon_2) \ldots (1-\epsilon_{n-1})\epsilon_n,$$

where ϵ_n is the probability that the game would terminate immediately after the *n*th move.

de Groot (1965, p. 25) mentions the analysis of one game in which the average number of good moves per position did not exceed 2, but he does not state what the average was. The analysis of several games would be required in order to obtain reliable estimates of the a_i 's(i=1, 2, 3, ...). For n>15, an adequate approximation to ϵ_n is 1/40, as far as the present analysis is concerned.

Another approach is to consider how many opening lines are recorded in Evans (1965). There are about ten thousand lines given to an average depth of about 12 moves, so the average number of moves considered in each position is about

$$10\ 000^{1/24} = 1.48.$$

These are almost entirely lines that have been played or analysed by good players. It seems a reasonable guess that

$$1 \cdot 6 < (1 - \epsilon_n)a_n < 1 \cdot 9$$

for most values of n, if extrapolation from the openings to the rest of the game is justifiable. (It is interesting to observe that, if 1.75 is the correct average in the opening, then *Modern Chess Openings* ought to run to fifty volumes, that is, only a fiftieth of equally good lines are at present recorded. This might give some measure of the conventionality of 'the book'.)

If this guess is right, then the number of games of not more than m moves on each side is roughly

$$\frac{1}{40} \cdot \frac{\mu^{2m+1}-1}{\mu-1}$$

where $1.6 < \mu < 1.9$, that is, the number lies between $1.6^{2m+1}/24$ and $1.9^{2m+1}/70$ These bounds are tabulated approximately in Table 1.

TABLE 1

The number of good games of not more than m moves on each side

| m | 10 | 20 | 30 | 40 | 50 | 100 | 200 |
|--|--------|-----------------|----------|-----------------------------|----------|-------------------|------------------------|
| log ₁₀ (lower bound) log ₁₀ (upper bound) | 3 4 | 7 9 <u>1</u> | 11 15 | 15 20 1 20 | 19 26 | 39 <u>1</u> 54 | 80] 110 |

It is interesting to note that, although some 99 per cent of games terminate before White's hundredth move, nevertheless the number of possible well-played games longer than this is far greater than the number that are shorter.

The number of good 'lines', that is, games not necessarily terminated, up to Black's *m*th move, is about $(\mu^{2m+1}-1)/(\mu-1)$ and is bounded by about $1\cdot 6^{2m+2}$ and $1\cdot 9^{2m+1}$. These bounds are tabulated in Table 2.

TABLE 2

The number of good 'lines' up to Black's mth move

| т | 10 | 20 | 30 | 40 | 50 | 100 | 200 |
|---------------------------------|------------|----------------|-------------|-----------------|----|-----|-----|
| log ₁₀ (lower bound) | 4 <u>1</u> | 8 1 | 12 <u>1</u> | 17 | 21 | 41 | 82 |
| log ₁₀ (upper bound) | 6 | 111 | 17 | 22 1 | 28 | 56 | 112 |

The above table can be inverted. For example, we can infer from it that a catalogue of 10^{12} good lines would extend to between 21 and 29 moves on each side, or say 25 moves, give or take a few moves. Moreover, if the machine had the white pieces, and was satisfied to adopt a fixed repertoire, 'in principle' it could get to about this depth with a catalogue of only a million lines (this assumes the square root law mentioned in Appendix J). As soon as the opponent played a slightly inferior move, the machine would have to begin exercising intelligence.

APPENDIX E. THE NUMBER OF POSSIBLE CHESS POSITIONS

Shannon (1950, p. 260) states that the number of possible chess positions is of the general order of $64!/(32!8!^22!^6)$ or roughly 10^{43} . The formula indicates that he

was assuming that no pawn had been promoted. In Good (1951), I calculated that the number of positions in which no pawn has been promoted, and there are no doubled pawns, is less than 2×10^{39} . The number of positions in which no capture has occurred is about 10^{32} . Allowing for all possibilities the number is less than

64.63 $\sum_{r=0}^{30} {62 \choose r} 10^r$ which is about 2×10^{50} , but if the number of moves since the last

capture or pawn move is regarded as part of the position, the upper bound is 10^{52} . I should say that 10^{46} is correct for the total, within a factor of a thousand. But most of these positions would be fabulously improbable. A master chess player would be happy if he knew what the best move was in 99.9 per cent of the positions, with nearly level material, weighted with their probabilities of occurring, that would occur in master chess, without blunders. Judging by the tables in Appendix D, and allowing for the possibility of repetitions of positions in the end-game, it may be that this number would not be more than about 10^{24} . Conceivably a good estimate of the total number could be made if a very comprehensive categorisation of chess positions were first produced, in which positions would fall into clumps of trillions of positions for the pieces, without pawns, and, for each of them estimate the number of ways of distributing the pawns legally—in effect a method of stratified sampling.

To make a rough comparison of the number of possible chess positions with that of draughts positions, the very crude upper bounds, '13⁶⁴ and 5³² or 3³², suggest that the former is about the cube or fourth power of the latter.

APPENDIX F. THE VALUE OF THE PIECES AND SQUARES

A satisfactory theory of the values of positions should include a theory for the values of the pieces as a special case. So it is worth while to look for a theory that gives reasonable values to the pieces. Then we could try to generalise the theory.

The values of the pieces P, N, B, R and Q, have been found by experience to be approximately proportional to 1, 3, 3, 5 and 9, and a king is worth about 4 in the end-game. These values vary with the position and with the number of pieces on the board. For example, two knights are worth less than rook when the only other pieces on the board are two kings, in so far as two knights and a king cannot force mate against a lone king. But the values given are applicable rather widely. For example, two knights are about the equal of six pawns on the second rank even when the kings are removed, as I have found by experiment.

A theoretical attempt to evaluate the pieces was made by H. M. Taylor in 1876, reported by Coxeter (1940, pp. 162-165). The value of a piece is taken as proportional to the average number of squares controlled, averaged over all 64 positions of the piece on the board. This argument leads to the relative values of N, B, R and Q: 3, 5, 8 and 13. Coxeter (or Taylor) goes on to modify the argument by asking instead for the probability of 'safely' giving check, that is, without being en prise to the king, if the piece and king are both placed on the board at random. This gives the ratios 12, 14, 22 and 40. These values are good, but this modification of the argument is artificial. For the piece values seem to be valid even for chess without kings, as I have found from some limited experiments, such as the one just mentioned. In fact they are probably more valid when there are no kings. Consider. for example, a piece, which I shall call the Galactic Emperor, that controls all the squares on the board. When the kings are on, it is worth at least seven queens, but, measured by squares dominated, it is worth only about three queens, on an open board. Therefore I believe that a theory that allows for the presence of the kings is too difficult to start with, and that it is more sensible to try first to obtain a theory that arrives at the known values without reference to checking.

In about 1936 (unpublished) I tried to make some allowance for the number of pieces on the board:

Let p be the probability that a square of the board is occupied. I assumed that the value of a piece is proportional to the expected number of squares dominated, averaged for all positions of the piece, and obtained the following values for the knight, bishop and rook.

$$E_{\rm R}(p) = 5.25 \text{ (independent of } p),$$

$$E_{\rm B}(p) = (140 - 196p + 182p^2 - 112p^3 + 44p^4 - 10p^5 + p^6)/16,$$

$$E_{\rm R}(p) = \frac{4}{p} - \frac{1}{2p} \left\{ \frac{1 - (1 - p)^8}{p} \right\}.$$

The expected number of squares dominated by the queen is $E_0(p) = E_R(p) + E_B(p)$. But we should allow also for the fact that rook and bishop have overlapping fields of influence. For example, a rook on Q5 and a bishop on KB4, on an open board, dominate only 24 squares instead of 27, and so lose about one ninth of their value. This gives a partial explanation of why a queen is worth more than a rook plus a bishop. A similar argument explains why two bishops are worth more than a bishop and knight, or than two knights. It is usually better to control two squares than to control one square twice, not allowing for the different values of squares.

Table 3 gives the numerical values of $E_{\rm R}(p)$, etc., for a few values of p.

TABLE 3 Values of $E_N(p)$, etc.

| р | 1 | ł | 1/3 | 1/4 | 0 |
|---|------|------|------|------|------|
| N | 5·25 | 5·25 | 5·25 | 5·25 | 5·25 |
| B | 3·07 | 4·7 | 5·72 | 6·35 | 8·75 |
| R | 3·5 | 6 | 7·5 | 9 | 14 |

The case p=1 cannot occur in ordinary chess, but it is certainly reasonable that a knight would be worth more than a rook if all 64 squares were occupied. The case $p=\frac{1}{2}$ is not expected to apply for the first several moves of a game, when we know roughly where the pieces stand. The initial position can be examined in its own right, as indeed can any given position. But it would be too short-sighted to say that rooks, bishops and queens have no value in the initial position, merely because they cannot move. It would be more reasonable to make some guess concerning the probability that each piece will occupy the various squares after any given number of moves, and then to use these probabilities as weights in a weighted average of the numbers of squares dominated. Moreover we should discount the future in some way, for example, by giving further weights of μ , μ^3 , ..., to positions 1, 2, 3, ... moves ahead, where μ is some positive number less than unity.

The case p = 1/3, which is typical in the middle-game, gives values that are generally regarded as good average values of the pieces. It would be interesting to try to verify the remainder of Table 3 experimentally.

To make the theory more accurate we may attach a value to each square, making the central squares more valuable than the others. To do this we can use a similar principle to the one used for pieces, that is, we can find the average number of squares dominated from the square when various pieces are placed there. These can be taken as 1Q, 2Rs, 1B (only one colour!), 2Ns, but not 8Ps since not every pawn has a reasonable chance of getting to the given square. It seems reasonable

to assume $1\frac{1}{2}$ pawns. I found the following values for the squares by this method. The squares are labelled as in Fig. 3.



FIG. 3. Labelling of squares. The labelling is centrosymmetrical, apart from the edges.

A: $87-84p+44p^2-8p^3$. B: $83-82p+55p^2-20p^3+3p^4$. C: $83-92p+80p^2-40p^3+8p^4$. D: $75-90p+95p^2-50p^3+18p^4-3p^5$. E: $75-100p+110p^2-70p^3+23p^4-15p^5$. F: $71-120p+160p^2-120p^3+48p^4-8p^5$. G: $65-108p+130p^2-110p^3+63p^4-21p^5+3p^6$. H: $65-118p+155p^2-130p^3+86p^4-21p^5+3p^6$. I: $63-138p+205p^2-180p^3+93p^4-26p^5+3p^6$.

We can use these values of the squares to improve the estimates for the values of the pieces. We could then use these improved estimates to improve the estimates for the squares and so on. There will always be an error because of the assumption that the density of p is the same over the whole board, and also because we can *choose*, for example, not to keep knights at the edge of the board, as a general rule. The chance that a knight is on the edge, after the opening, is much less than 28/64, if the players are any good!

I did this algebra before there were any electronic computers. It will now be easy to do the arithmetic.

If we wish to allow for the existence of kings we should give extra weight to the squares in their vicinities, or in some other way. We might, for example, compute a 'checking value' for each piece as the probability of giving check without being *en prise* to the king, and then take a linear combination of the checking value and the square-domination value.

The values of the pawns require a different theory since (i) they can be promoted, and (ii) they move differently from the way they capture.

It should be considered whether the values of the pieces can be related to the probability of winning. Should the value be taken as the improvement in the log-odds of winning as against losing?

Theories of evaluation of pieces could be tested by inventing new pieces; we are not inevitably tied to the small sample of distinct kinds of piece that occur in ordinary chess. An example of this test, was given above in connection with the Galactic Emperor, although perhaps this piece is too extreme for most purposes.

APPENDIX G. CATEGORISATION OF MATING POSITIONS AND CATALOGUE OF FREQUENT MATING MOTIFS

When attacking the king, human players find it useful to be able to recognise the possibility of mating motifs and mating positions. To have a repertoire of these

things is helpful in goal formation and in the pollarding or pruning of analysis trees.

For mating motifs, a useful reference is Vukovic (1965), and there are many other books that deal with this topic. Here I should like to mention a categorisation of mating positions.

(a) *Three-filer*. An example is a mate with two rooks, the opponent's king being on the edge of the board. It is natural to think of the three squares off the board as forming an imaginary file. Another example is mate with a rook and king against a king on the edge, or a mate on the back line, when the opponent's king has three pawns in front of it.

(b) T and tunnel. For example, the final position in the game 1 P-K4, P-K4; 2 Q-R5, K-K2; 3 Q×KP mate.

v

(c) Cross diagonals and L. For example,

| | | | | • | • | ٠ | • | ٠ | ٠ | ٠ | L. |
|---|---|---|-----------|---|---|---|---|---|---|---|----|
| | | | N R. | | | | B | | | Q | |
| • | | • | . p k . P | • | | | • | • | k | | |
| • | • | • | p | | • | | | | • | • | |

(d) Parallel diagonals and two stops. For example,

| • | ٠ | ĸ | r | • | • | • | • | |
|---|---|---|---|---|---|---|---|--|
| • | р | ٠ | • | • | • | • | • | |
| N | N | • | | | | • | | |

With a little artificiality, most mating positions can be put into one or more of these categories. If a machine could do this it would really have arrived.

APPENDIX H. QUIESCENCE, TURBULENCE, AND AGITATION

Positions for which there is no move forceful or purposeful enough to be worth analysing were independently described as 'quiescent' by Shannon (1950) and by Good (1950b). Turing (1953) called such positions 'dead'. One objection to this term is that a 'dead draw' is a standard expression among chess players. Also it is useful to be able to refer to 'degrees of quiescence', whereas 'degrees of deadness' sounds facetious, in spite of the expression 'dead as a door-nail' and of the notorious variability of deadlines. A good name for the degree of quiescence is simply the 'quiescence' of the position, or inversely one might refer to the 'turbulence'. A quiescent position is one of low turbulence, a combinative position is one of high turbulence. It is possible to define turbulence with respect to goals other than material balance, such as control of the centre (cf. Newell *et al.* 1958). The position illustrated in Fig. 4 is of high turbulence with respect to material balance.

| • | • | ٠ | • | • | • | • | • | |
|---|---|---|---|---|---|---|---|--|
| • | | k | n | R | | | | |
| • | • | • | r | Q | b | В | | |
| р | р | р | b | q | n | р | р | |
| Р | Р | Р | R | Ν | Ν | Р | Р | |
| • | | | р | р | r | В | | |
| | | | Р | Р | Р | | • | |
| • | • | • | • | • | Κ | • | • | |

FIG. 4. A position of high turbulence!

Before considering a more precise definition of 'turbulence' we discuss an allied notion called 'agitation'.

If, in the analysis of a position π_0 we reach a position π , the turbulence of π is not quite an adequate criterion for deciding whether to suspend further analysis. If π were obviously won we would suspend its analysis even if it were turbulent. We need some measure, called say 'agitation', that measures turbulence and also takes into account whether the position is obviously won, drawn, or lost. If we can measure the agitation of π , then we would set a threshold on this agitation multiplied by $P(\pi | \pi_0)$, the probability of reaching π from the current position π_0 . (Notice that, when we estimate this probability, we are in part, estimating the probability of our own future actions, not just those of our opponent. In my opinion, the same thing happens in the definition of a 'decision'. See Good 1964.) If this threshold is not reached, we would almost suspend analysis of π . I say 'almost' because there are a few points to attend to before finally classifying π as an endpoint on the analysis tree of π_0 . We should first (i) check whether π is a position known definitely to be won, drawn, or lost; (ii) if π is favourable to us, we should see if we have a move that prevents check; (iii) if π is unfavourable to us, we should see if we have a check; (iv) if π seems to be an obvious draw, we should carry out both the tests (ii) and (iii).

How then should 'agitation' be measured? It will be recalled from Appendix C that the utility of position π , if we reach it, is of the form $P(W|\pi) - \lambda P(L|\pi)$, where $P(W|\pi)$ and $P(L|\pi)$ are, in some sense, the probabilities of a win or loss, given π . If we decide to treat π as an endpoint, then these probabilities are 'superficial' and $P(W|\pi) - \lambda P(L|\pi)$ may reasonably be described as the 'superficial expectation of the utility' or, for short, as the 'superficial utility'. If we decide to suspend the analysis of π , that is, to regard it as an endpoint, it should be because we believe that a deeper estimate of the utility of π , obtained say by spending a dollar on its analysis, is unlikely to make much difference. We must therefore have some method of guessing the expected value of $|U(\pi|\$) - U(\pi)|$, where $U(\pi)$ is the superficial utility of π , and $U(\pi | \$)$ is the expected utility of π in the light of a dollar's worth of analysis, \$, or of course some other unit. The agitation $A(\pi) = A(\pi | \$)$, of π is the guessed expected value of $|U(\pi|\$) - U(\pi)|$, and the decision whether to treat π as an endpoint in the analysis of π_0 is to be made by setting a threshold on $A(\pi)P(\pi|\pi_0)$. Part of the five-year plan is to find an evaluation function that estimates $U(\pi)$, and another one that estimates $A(\pi | \$)$. We need evaluation functions even for turbulent positions.

A well-known issue in the theory of rationality is whether the distribution of a utility is relevant except in regard to its expectation. My own belief, shared by most 'rationalists', is that only the expectation is relevant, but this is on the assumption that all the relevant calculations have been completed. In complicated problems, like mathematics and chess, this assumption is unrealistic, and superficial probabilities and expectations have to be used (cf. Good 1962b). The guessed expected value of $|U(\pi|) - U(\pi)|$ is roughly proportional to the standard deviation of the superficial subjective distribution of the true utility of π . The true utility of π is of course either 1 or $-\lambda$, or $\frac{1}{2}(1-\lambda)$, if both we and our opponent are perfectly rational.

I believe that this notion of the agitation of a position is applicable to almost every decision in practical life, since it is very rare that we can complete our reasoning in a practical problem. Instead of using the axiom of probability that P(E|H) = P(F|H) when E and F are logically equivalent, we are forced to use the weaker one that these probabilities are equal when we have *proved* that E is equivalent to F (Good 1950a, p. 49.) Both humans and machines must often use imperfect logic.

It seems reasonable to define the turbulence, $\tau(\pi|G)$, of π with respect to some goal G, as the superficial expectation, or guessed value, of $|L(G|\pi, \$) - L(G|\pi)|$, which equals $|W(G:\$|\pi)|$, where L denotes log-odds, W denotes 'weight of evidence', and the colon denotes 'provided by'. But it seems that decisions should depend

114

more on agitation than on turbulence. The utilities should allow for all goals that we are prepared to consider. Agitation with respect to a goal can also be defined by interpreting U appropriately.

If these ideas could be properly worked out, they would be applicable to the question of finding the value of storing previously analysed positions, as in Samuel's draughts program.

At the end of Section 1, we mentioned the problem of deciding what moves to discard 'without analysis'. I think the right approach to this question is to try the move, and estimate $|U(\pi')-U(\pi)|P(\pi|\pi_0)$ where π' is the position reached after that move is played, and the utility $U(\pi')$ is estimated on the (false) assumption that the player retains the move. If this estimate is small enough, then the move can be treated as of no importance. In fact, if this estimate is small, then $P(\pi'|\pi)$ would be small, since a free move is usually quite advantageous. An exception is when the player is *Zugzwang*, and also in some end-game positions when no move is any good.

APPENDIX J. HOW SHOULD THE OPENINGS BE PROGRAMMED? SHOULD CHESS BE REPLACED BY RANDOMISED CHESS?

One obvious suggestion for handling many of the openings is to store nearly the whole of *Modern Chess Openings*. Then the machine will not go badly wrong until the opponent leaves the book. Moreover, some of the published lines would be clear wins for the machine, whereas those that were known to be bad for the machine could be avoided. In correspondence chess, this device would put the machine at a disadvantage, but in 'over the board' chess, the machine would sometimes win without any 'intellectual effort'. This would indeed be a hollow victory.*

Chess was presumably invented as a test of a kind of intelligence, and it is no more reasonable that the machine should be allowed to win by throwing the whole book at its opponent than that humans should be allowed to win tournaments by the learning of innumerable lines. Both for machine programming, and for the health of chess in general, the present rules of the game should undoubtedly be modified, and so-called 'randomised chess' should be adopted by the International Chess Federation, the F.I.D.E. (see, for example, Good 1952-55, 1965, p. 35).

In randomised chess the white pieces on the back line are randomly permuted before the game begins, but if the two bishops turn out to be on squares of the same colour the one on the right is interchanged with the piece on its left. This gives rise to 1440 essentially distinct positions if the black pieces are set up by mirror reflection. Castling can be defined by the rule that the rook is moved up to the king and the king hopped over it. Randomised chess has exactly the same seriousness as ordinary chess, and the basic principles, such as the need to control the centre, are the same. The theory of the openings is genuine theory and not largely practice and fashion as in ordinary chess.

But since it is too much to hope that the F.I.D.E. will be rational enough to organise this change in the face of opposition from vested interests, let us consider what we can do other than swallowing the whole book.

The chess openings form a very large tree: there are about 150 000 moves in *Modern Chess Openings*, consisting of some 10 000 lines. But a player with the white pieces, and a fixed repertoire, need learn only about 100 lines, the square root of the number in the book, and another 100 lines for when he has the black pieces. The argument for this, which is not at all rigorous, is that the number of lines in the book is roughly k^n , where k is the number of choices in each position, and n is the length of the line; whereas a repertoire for White need have only about $k^{n/2}$ lines, since he can make a fixed choice in each book position. The argument

* It would not be hollow if it provoked the F.I.D.E. to go over to Randomised Chess.

is not rigorous because k is a decreasing function of the number of moves from the initial position, since the book is intended to be a book about the *openings*.

At any rate, one way that a computer could be used to help a chess player would be in the selection of an opening repertoire for him. (A repertoire is a Borel-von Neumann pure 'strategy' or 'play', but is incomplete.) If the repertoire were selected randomly, then the player's opponents would not be in a position for some time to take advantage of his published games in order to prepare variations against him. Likewise the machine could select a repertoire for say its next ten games, It would then be able to start a game quickly 'over the board', and perhaps finish by correspondence.

The problem of selecting an opening repertoire or subtree can be framed in several ways. For example, we could try to find a repertoire having minimum storage requirements (which would be especially useful for human players), or one with a preference for open games. It might well turn out that a machine would play a better combinative than strategic game in comparison with humans.

APPENDIX K. SYNERGY, A POINT OF TERMINOLOGY

Close interaction between men and machines is sometimes called 'man-machine symbiosis' and sometimes 'man-machine interaction'. The first term is inappropriate since, in correct usage, it refers to purely biological interaction. The second term does not convey the notion of the closeness of the interaction. Fortunately there is an English word already available, it is 'synergy'. In my opinion the 1970s will be remembered as the age of synergetics.

REFERENCES

Bernstein, A. et al. (1958a). A chess-playing program for the IBM 704 computer. Proc. Western Jt Computer Conf. (WJCC), 157-159.

Bernstein, A., & Roberts, M. de V. (1958b). Computer vs. chess-player. Scient. American, 198, 96-105.

Borel, E., Fréchet, M., & von Neumann, J. (1953). Discussion of the early history of the theory of games, with special reference to the minimax theorem. *Econometrica*, 21, 97-127.

Coxeter, H. S. M. (1940). *Mathematical Recreations and Essays*, from the original by W. W. Rouse Ball. London: Macmillan.

Euwe, M., & Kramer, H. (1964-65). The Middle Game, two volumes. London: Bell.

Evans, Larry (1965). *Modern Chess Openings* (Griffith and White), 10th edn. W. Korn (ed.). London: Pitman.

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

Golombek, H. (1966). Luck in chess. Observer Weekend Review, January 23, p. 31. London.

Good, I. J. (1939). Mathematics and chess, Eureka, 1, 9-11.

Good, I. J. (1949). The number of individuals in a casade process, *Proc. Cambridge Philosophical Soc.*, **45**, 360-363.

Good, I. J. (1950a). *Probability and the Weighing of Evidence*. London: Griffin; New York: Hafner.

Good, I. J. (1950b). Contribution to the discussion on Slater (1950), p. 199.

Good, I. J. (1951). Private communication to Mr Leon Jacobson.

116

- Good, I. J. (1952/55). Notes on Randomised Chess in Chess, 17, 161, 1952; 20, 203, 1955; Brit. Chess Mag., 73, 213-214, 1953.
- Good, I. J. (1954). Contribution to the discussion of a paper by J. M. Hammersley and K. W. Morton. J. Roy. Stat. Soc. B, 16, 68.
- Good, I. J. (1957). Letter in Chess, 22, 293-294, containing misprints, and concerning a form of three-dimensional chess played on a $4 \times 4 \times 6$ 'board'.
- Good, I. J. (1959). Could a machine make probability judgments? Computers and Automation, 8, 14-16, 24-26.
- Good, I. J. (1962a). Botryological speculations, *The Scientist Speculates*, I. J.
 Good, A. J. Mayne, J. Maynard Smith (eds.), 120-132. London: Heinemann;
 New York: Basic Books.
- Good, I. J. (1962b). How rational should a manager be? Management Sc. 8, 383-393; and in Executive Readings in Management Science, M. K. Starr (ed.). New York: Macmillan (1965).
- Good, I. J. (1964). Measurement of decisions, New Perspectives in Organisation Research, pp. 391-404. W. W. Cooper, J. Leavitt, and M. W. Shelly II (eds.). New York: Wiley.
- Good, I. J. (1965a). Speculations concerning the first ultraintelligent machine, Advances in Computers, Vol. 6, pp. 31-88. F. L. Alt and M. Rubinoff (eds.). New York and London: Academic Press.
- Good, I. J. (1965b). 'The mystery of Go.' New Scientist, 25, 172-174.
- Groot, Adriaan D. de (1965). Thought and Choice in Chess, G. W. Baylor (ed.). The Hague and Paris: Mouton. A translation, with additions, of a Dutch version of 1946.
- Horowitz, I. A., & Mott-Smith, G. (1960). Point Count Chess. New York: Simon and Schuster.
- Kister, J., Stein, P., Ulam, S., Walden, W., & Wells, M. (1957). Experiments in chess. J. Assn Comput. Mach., 4, 174-177.
- Lasker, Edward (1951). Modern Chess Strategy. London: Bell.
- Lasker, Emanuel (1932). Lasker's Manual of Chess, 2nd ed. London: Constable; Dover reprint, 1960.
- Luce, R. D., & Raiffa, H. (1957). Games and Decisions. New York: Wiley.
- Macdonald, D. K. C. (1950). Contribution to the discussion on Slater (1950), p. 198.
- Maynard Smith, J., & Michie, D. (1961). Machines that play games. New Scientist, 12, 367-369.
- McCulloch, W. (1965). Embodiments of Mind. Cambridge, Massachusetts: M.I.T. Press.
- Michie, D. (1966). 'Game-playing and game-learning automata', with an appendix by J. Maynard Smith; in Advances in Programming and Non-Numerical Computation, pp. 183-200. Oxford, etc.: Pergamon Press.
- Neumann, J. von, & Morgenstern, O. (1944-47). The Theory of Games and Economic Behaviour. Princeton: University Press.
- Newell, A., Shaw, J. C., & Simon, H. A. (1958, 1963). Chess-playing programs and the problem of complexity. *IBM J. Res. Dev.* 2, 320-335, 1958. Reprinted in *Computers and Thought* (see Samuel 1963), 39-70.
- Newell, A., & Simon, H. A. (1961). Computer simulation of human thinking. *Science*, 134, 2011-2017.
- Newell, A., & Simon, H. A. (1965). An example of human chess play in the light of chess playing programs. *Progress in Biocybernetics*, pp. 19-65. Amsterdam, London, New York: Elsevier.
- Philidor, F. A. D. (1749). L'Analyse des Échecs. London.
- Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM J. Res. Dev.*, **3**, 210-229.

Samuel, A. L. (1963). Reprint of Samuel (1959), with an additional appendix. In *Computers and Thought*, E. A. Feigenbaum and J. Feldman (eds.), pp. 71-105. New York, etc.: McGraw-Hill.

Shannon, C. E. (1950). Programming a computer for playing chess. Phil. Mag. (London) 7th Ser., 41, 256-275.

Slater, Eliot (1950). Statistics for the chess computer and the factor of mobility. Symposium on Information Theory, pp. 150-152. London: Ministry of Supply.

Suetin, A. S. (1965). *Modern Chess Opening Theory* (tr. from the Russian by D. J. Richards, ed. by P. H. Clarke. Oxford, etc.: Pergamon; date of the original Russian suppressed).

Turing, A. M. (1953). Digital computers applied to games, *Faster than Thought*, B. V. Bowden (ed.), pp. 286-295. London: Pitman.

Vukovic, V. (1965). The Art of Attack in Chess (tr. by A. F. Bottrall, ed. by P. M. Clarke. Oxford, etc.: Pergamon.