Man vs. Machine: The Rivalry in Chess

James Brofos

July 26, 2013

Abstract

The game of chess is captivating for its positional complexity, and impressive for the mental determination required to play well. It is also the game that presents a non-trivial challenge to computer scientists to create powerful "chess engines" to demolish human opponents in spectacular exhibition matches. The history and future of these chess engines are the topic of this paper, and the problems typical of chess in general, and of computer chess in particular, are offered "for fun" to the readership.

1 A Brief History of (Computer) Chess

> I'm sorry Frank, I think you missed it. Queen to bishop three, bishop takes queen, knight takes bishop, mate.
>
> – HAL 9000 in *2001: A Space Odyssey* to Frank Poole[1]

These days, nearly everyone has had exposure to the game of chess. The players themselves may have contended from across the board or looked on as others played. They may have attended the meetings of the local chess club or competed in tournaments. Some may have even aspired to the titled ranks of the *Fédération Internationale des Échecs* (abbr. FIDE, or World Chess Federation). The appeal of chess has a legacy millennia long, springing into existence first in northern regions of Afghanistan or India around 500 AD [23]. In all this time there have been landmark events which have influenced the game in the highest degree. Examples of these include: in the 15$^{\text{th}}$ Century the emergence of modern chess in Italy [14]; in 1656 the first recorded chess analyses by Gioachino Greco [8]; in 1948 Mikhail Botvinnik's victory in the FIDE World Championship to usher in an era of Soviet supremacy; in 1972 the Cold War confrontation between American Bobby Fischer and Soviet Boris Spassky in the *Match of the Century* in Reykjavík, Iceland; in 1996 the games between Garry Kasparov and the IBM supercomputer *Deep Blue* with a convincing three victories, two draws, and one loss in humanity's favor.

1.1 The Opening Game: Prior to Deep Blue

The Deep Blue-Kasparov match, and the subsequent rematch in 1997, represent something of a culmination in the history of computers in chess. And it is to this fragment in the history of the game that we devote our attention. The novelty of a chess-playing automaton emerged for the first time in 18$^{\text{th}}$ Century Hungary under the auspices of diplomat-inventor Wolfgang von Kempelen [47]. Von Kempelen's machine, affectionately called "The Turk" by its contemporaries, was intended to amuse the courts of Napoleon and Maria-Theresa. Even Benjamin Franklin, a comparatively able player for his day, was extended a personal invitation to match wits against The Turk:

> If I have not, immediately upon my return from Versailles, renewed my request that you will be present at a performance of my automaton chess player, it was only to gain a few days in which I might make some progress in another very interesting machine, upon which I have

been employed and which I wish you to see at the same time. Please sir, have the kindness to inform me of the day and hour when I shall have the honor of receiving you in my rooms.
– von Kempelen to Benjamin Franklin, May 28$^{\text{th}}$, 1783 [19]

Yet, as shall be seen, von Kempelen's contraption must have been, from a computational perspective, something of a "let-down" for the field of automata in chess. The apparent appeal of The Turk, and the subsequent source of disappointment, was found in its presentation of the *illusion* of mechanical thought.

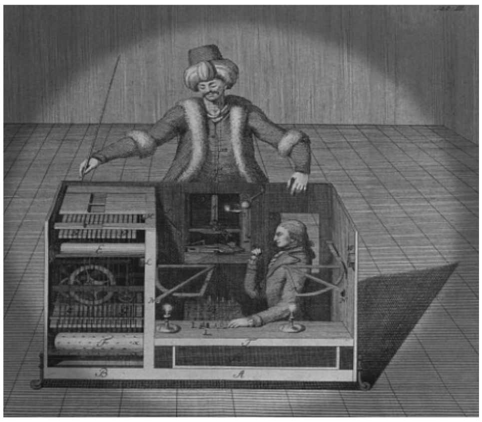


Figure 1: Von Kempelen's fantastic illusion revealed as a hoax. Witnesses to The Turk's apparent "skill" attested that, during games, there was no communication of the opponent's moves – so, in other words, a human operator of the machine would have no knowledge of the board after the first turn. The solution to this conundrum was unveiled to be a clever system of magnets hidden beneath the machine that informed the operator of where the opponent had played in the previous move [27, 22].

In actuality, von Kempelen's machine was just an elaborate deception. Yet, for all that, The Turk had introduced, for the first time in history, the idea that a *machine* could engage in, and succeed at, a distinctly *human* game. Further, it constituted an instance which elevated the quiet game of chess to something of a celebrity. Benjamin Franklin himself, as asserted by his grandson, "enjoyed his game with the automaton and was pleased with it" [22]. Edgar Allan Poe in 1836 wrote the essay "Maelzel's Chess-Player"[1] in speculation of the device's *modus operandi*, declaring in the first line of his exposition, "Perhaps no exhibition of the kind has ever elicited so general attention as the Chess-Player of Maelzel. Wherever seen it has been an object of intense curiosity, to all persons who think" [26].

Yet after the spectacular success of The Turk in the 18$^{\text{th}}$ and 19$^{\text{th}}$ Centuries, the public's fascination with "thinking machines" in chess plunged into an abyss for more than one-hundred years. Indeed, it would require the onset of the Second World War and the advent of early computer technology to reignite interest in chess automata. Enter Claude Shannon, a computer scientist then in the employ of Bell Telephone Laboratories. In 1949, Shannon published his revolutionary "Programming a Computer for Playing Chess" [30]. The work proposed the first algorithm to empower a computer with the ability to intelligently play chess, which relied on a theoretical board evaluation function $f(P)$, where

$$f\colon \{\text{board space}\} \to \mathbf{R}$$

and so P represents the state of the board at any given turn and the resulting real number the strength of the board. Shannon's idea is relatively intuitive and elegant: he asserted that every position that can legally arise in a game of chess may be numerically captured by attributes that are characteristic of a strong or weak position for the player, and from an evaluation of those attributes can spring all manner

[1]Poe refers to the device as "Maelzel's chess-player" after The Turk toured the United States with Germany's Johann Mälzel. Mälzel delivered The Turk to the shores of the US in 1825, twenty-one years after von Kempelen's death in Vienna.

of strategy and foresight required for skillful play. Positions that were advantageous for the computer player would be evaluated positively with higher numbers reflecting stronger winning chances, while losing positions would be evaluated negatively. Games where the position was drawn would be evaluated as zero. On the basis of centuries-worth of human experience in determining the relative strength of a position, Shannon identified the following as being crucial to any evaluation function $f(P)$:

1. Piece values: According to the three-time United States Open Champion I. A. Horowitz, every piece in chess has a strength relative to the worth of a single Pawn. Thus, if a Pawn represents a single unit of strength, then, generally speaking, the Knight is equivalent to three Pawns, the Bishop three Pawns and some slight amount more, the Rook five Pawns, and the Queen nine, while the King is infinitely valuable (since a loss of the king means a lost game) [12]. The side possessing the greater material advantage according to this metric has some advantage in the position.
2. King safety: A King that is vulnerable to attack by the opponent reflects fragility in the position. The side whose King is well protected (e.g. by castling) when the opponent's King is relatively undefended, enjoys benefit in the position. See Figure 2.b.
3. Pawn structure: A poor Pawn structure is characteristic of a weak position. In chess the Pawn structure is weakened by the existence of isolated Pawns (Pawns that are unprotected by supporting Pawns), doubled Pawns (Pawns that, by capturing a piece, have wound up on the same column of the board), and backward Pawns (Pawns that support other Pawns, but are incapable of advancing due to lack of protection). Lack of sound Pawn structure allows for easy infiltration of the camp by the enemy, which is an advantage to the opponent. See Figure 2.c.
4. Piece development: Shannon characterizes this attribute of a position by the example that strong Rooks are typically placed on the open rows or columns (i.e., no piece inhibits their range of movement along that dimension) of the board. The key idea here is that pieces should be moved out of their initial positions on the board to squares where their attacking chances are best put to use. Especially in the opening phase of chess, the side with superior development will typically have the better game.

(a) The opening position of chess from which all board variations must inevitably spring. Game sequences exist in finite number, yet are so multitudinous as to defy enumeration by computer.

(b) White has sacrificed a piece to obtain a dangerous attack on the Black sovereign. Thus, Black's material advantage in this position may be sorely outweighed by the relative insecurity of his King.

(c) The Black Queen-side Pawns cannot defend themselves against a White attack. Both sides have a position development, yet Black's pieces may be tied down in defense of the unstable flank.

Figure 2: Chess positions that are instructive both of the game itself and of the value of Shannon's key positional attributes. Chess diagrams adapted from those discussed first by Horowitz [12].

All chess diagrams created by the Shredder Classic 4 GUI [24].

Claude Shannon had set the stage for computer chess, and let loose on hordes of eager mathematicians a complex problem in desperate need of a solution. Shannon himself considered the computational puzzle of profound significance to the field of computer science, claiming that a solution to computer chess would be the latchkey to "Machines capable of translating from one language to another[, of] making strategic decisions in simplified military operations [. . .] Machines capable of orchestrating a melody[,] capable of logical deduction" [30]. In the years following 1949, numerous attempts were made to create computer chess players. In 1951, for instance, the German-born Dietrich Prinz developed at the University of Manchester a program capable of solving chess' "mate-in-two" problem[2] [5, 47]. See Figure 3.a. Yet Prinz' program was weak in the sense that it was restricted to endgame play (in fact, only the *last two moves* of the endgame).

(a) Magnus Carlsen vs Helgi Gretarsson, Rethymnon, 2003[3]

Carlsen won the game with White pieces with a mate-in-two from this position. If White were to play, Prinz' program could find the winning sequence of moves. Can you?[4]

(b) Bobby Fischer vs. Greenblatt Program, Cambridge, 1977

The final position of one of three games played between the former FIDE World Champion Bobby Fischer and *Greenblatt*, playing White and Black respectively.

Figure 3: Endgame positions in chess.

Other pioneers in computer scientists followed in Shannon's wake, including Alan Turing and Alex Bernstein of IBM. Bernstein's involvement in 1957 marked the dawn of two major events in computer chess history: first, IBM's first foray into the field of chess automata, initiating a four-decades-long journey that would culminate in the creation of Deep Blue; second, Bernstein and colleagues at MIT created the first computer program capable of playing a full game of chess against a human opponent [38]. Bernstein's chess program represented a vast improvement over Prinz' two-move solver in extent of ability, but still failed spectacularly to match the skills and intuition of a master human player. Indeed, International Master Edward Lasker delivered a crushing defeat to the IBM machine, remarking in victory that the machine had played perhaps a "passable amateur game" [48]. Yet the computational impasse confronting Bernstein and others was two-fold, as computers were falling short both in terms of strategic capability and in terms of speed. Prinz' computer required approximately fifteen minutes to solve a single mate-in-two puzzle, and Bernstein's program suffered eight minutes to evaluate a single position. Even if computers *could* outplay humans skillfully, they would certainly lose tournament games to time constraints. Computers needed to play well, and they needed to play quickly.

Hindered by the slow execution time of 1950s-era computers which were capable of only 10,000 calculations per second, chess pioneers turned to heuristics to improve machine understanding of positional strength. The idea of the heuristic emerged with MIT's Richard Greenblatt and his chess machine *Greenblatt*. What made *Greenblatt* unique among other Shannon-inspired chess automata was that it incorporated "considerable chess knowledge[, which had] been programmed in" [9]. If computers could not be made to evaluate positions *faster* they could as compensation be made to evaluate them *better*. Richard Greenblatt captured, in code, fifty aspects of the game that he had gleaned from experience and instinct, aspects that were beyond the scope of what might be grasped by Shannon's evaluation function $f(P)$. As a result of these improvements, *Greenblatt* was far and away the strongest chess automata of its day. But even so, the computer failed miserably to win even a single match against Bobby Fischer in a three game exhibition in 1977. See Figure 3.b.

Despite the setback, 1977 was a momentous year for computer chess if only on the grounds that the subject had emerged both as a legitimate computer science benchmark and also as something of a

[2]Chess' mate-in-two problem is a simple type of chess puzzle that challenges the player to find the two moves that, given any response from the opponent, will guarantee checkmate with correct play.

[3]I take the following as the convention to cite chess games: [The White Player] vs [The Black Player], [Site of Game], [Year of Game]. This is a fairly typical method for identifying chess games in databases.

[4]Answers to all chess exercises in this paper can be found in the appropriately-labeled section at the conclusion.

competitive sport. As if to capture the spirit of the times, Monty Newborn issued the following declaration at the Second Computer Chess Championship:

> In the past Grandmasters came to our computer tournaments to laugh. Today they come to watch. Soon they will come to learn.
> – Monty Newborn, Toronto, 1977 [45]

In the same year, Ken Thompson and Joe Condon, working at Bell Laboratories, recognized that the truly exceptional chess computers would be constructed not only on purpose-built software, but also on purpose-built hardware [47]. Thompson and Condon's machine, affectionately (and perhaps chauvinistically) called *Belle*, was assembled to contain circuitry that allowed for exceptionally fast positional evaluations. Belle was a realization that perhaps the singular advantage the computer had over the human mind was its ability to repetitively perform simple tasks very quickly and typically without severe error – Belle would use brute force, rather than intrinsic knowledge, to examine a massive number of possible variations (160,000 positions per second) that could emerge from the game [47].

All of the advancement in computer chess, and Monty Newborn's challenging statement, did not go unnoticed by the human chess community. In a bet made in 1968, Scottish International Master David Levy had boasted that no computer would be able to defeat him in ten years time [20]. Prior to the match, Levy had written that,

> Clearly, I shall win my [bet], and I would still win if the period were to be extended for another ten years. Prompted by the lack of conceptual progress over more than two decades, I am tempted to speculate that a computer program will not gain the title of International Master before the turn of the century and that the idea of an electronic world champion belongs only in the pages of a science fiction book [21].

Playing against the imaginatively-named chess automata *Chess* 4.7 – a machine constructed by Larry Atkin and David Slate of Northwestern University – fate had it that Levy would win his bet. In a match of six games, Levy claimed victory in four and fought one to a draw. Yet the Scotsman emerged from the ordeal somewhat humbled by the experience, no longer singing praises to the invincibility of the human world champion. Indeed, Levy confessed that the computer opponent was "very, very much stronger than I had thought possible when I started the bet" and that "Now nothing would surprise me" [20]. As shall be seen next, Levy's admissions were the harbingers of a kind of "fall from grace" for human chess champions.

1.2 Deep Blue vs Kasparov

In the mid-1990s, IBM's research division set out with the purpose of building "a world-class chess machine" that could play skillfully enough to defeat the most capable human opponent [6]. The matter of who represented the most capable human opponent was, at the time, not as contentious as it might have been. Garry Kasparov of Russia was the reigning FIDE World Chess Champion for the last fifteen years of the 20^{th} Century, having defeated Anatoly Karpov, the previous champion, in 1985. Further, Kasparov was widely regarded as one of the strongest chess players, if not *the* strongest, in the history of the game, having dominated the field of chess in the nineteen years between 1986 and 2005 as the world's highest rated player. Indeed, Kasparov also went on to achieve the highest rating in the history of the game, a record that was only recently outdone in January 2013 by Magnus Carlsen of Norway.[5] To challenge Kasparov's throne, IBM created the supercomputer Deep Blue.

Deep Blue began first in 1985 as the dissertation of Fen-hsiung Hsu, then a graduate student at Carnegie Mellon University. In its early years, Deep Blue was known by another name entirely: ChipTest [40]. Hsu and his classmate Murray Campbell worked as the chief architects of ChipTest's evaluation function, and the automata itself proved capable enough to win the 1987 North American Computer Chess Championship [41, 13]. The success of the university project attracted the attention of IBM's research division and the two were brought into the company with the extraordinary purpose of constructing a computer capable of defeating the World Chess Champion. Renaming the IBM project to "Deep Blue," a name inspired by an earlier incarnation of the program "Deep Thought" (itself named after Douglas Adams' famous computer character in *The Hitchhiker's Guide to the Galaxy* [18]), the IBM began the process of building its world-class machine.

[5] Chess players, as in many other competitive games, are ranked by the Elo system, which is designed to compare relative strength. The casual player is ranked at 1200 Elo, while the grandmaster level will typically start at 2500 Elo. Garry Kasparov achieved 2851 Elo, and Magnus Carlsen usurped the record with a performance of 2857 Elo.

Unfortunately for computer chess, Hsu's and Campbell's first forays against the world champion were met with defeat. The initial attempt to dethrone Kasparov came in 1989 with a two-game exhibition match played in New York. Kasparov, who had demonstrated his prowess against chess automata four years prior by demolishing thirty-two similar programs *simultaneously* in an exhibition match in Hamburg, Germany [16], was undoubtedly confident in his chances of victory. Like the machines that had come before it, IBM's chess playing contraption met with a quick end against the seemingly-insurmountable Russian King of Chess. In the face of the defeat Hsu and Campbell reconsidered their approach to the problem and realized that, while neither of them were unfamiliar to the game, they did not have the knowledge and skill possessed by chess masters at the *highest* level. So entered the American Grandmaster Joel Benjamin, brought to the IBM team explicitly as Deep Blue's "official Grandmaster consultant," training "the computer to think more positionally" [41, 44].

Development of Deep Blue continued to the year 1996, having been coached and adjusted and maintained for slightly over a decade since its inception. In the judgment of Hsu and Campbell, Deep Blue was ready for its first rematch against Kasparov. In a match arranged by the Association for Computing Machinery in Philadelphia, Deep Blue and Kasparov again contended from across the board − this time for a $500,000 prize [42]. The 1996 showdown did not go unnoticed by humanity, and the publicity storm around the event led to *Time Magazine* running the rather provocative cover story, "Can Machines Think?" [37]. Indeed, the Deep Blue-Kasparov match seemed to represent something of an "identity crisis," an apparent struggle for "the meaning and dignity of human life" [37]. If anyone had been anxious, then their fears were realized when the first game went to the IBM computer, marking the first time in history that a computer had unseated the reigning world champion in tournament play [46].

Yet after an undoubtedly grim start for humanity, Kasparov proved a resilient opponent, drawing the third and fourth games, while outright winning the second, fifth, and sixth, finally achieving a score of four against two[6] in Kasparov's favor. After the match, the victor remarked of Deep Blue that, "I could feel − I could smell − a new kind of intelligence across the table . . . Although I think I did see some signs of intelligence, it's a weird kind, an inefficient, inflexible kind that makes me think I have a few years left" [33]. Public interest would not be satisfied with letting matters stand, and another confrontation between the machine and the champion was set for 1997.

Played in New York City at the Equitable Center, IBM's revenge match in 1997 was fought out "with cameras running, press in attendance and millions watching the outcome" and called the "most spectacular chess event in history" [40, 7]. See Figure 4. Deep Blue, operating on 480 processors (versus a mere 216 a year prior) [6], had been vastly improved since its last encounter with the human world champion with Kasparov admitting that he was "afraid of the machine" [7]. Six games were again played between man and machine in "a small television studio. The audience watched the match on television screens in a basement theater in the building, several floors below where the match was actually held" [40]. This time, after the dust had settled, the computer emerged with two victories, Kasparov with one, and both with three drawn games. The victory represented the culmination of a three-centuries-old idea: that a computer could be made to play chess skillfully. In a legacy began with von Kempelen's The Turk and in a concept refined by Shannon and in an unrivaled implementation by Campbell and Hsu, Deep Blue had shown that computers were capable of making informed decisions in games involving strategy, forethought, and deep complexity. Today, Deep Blue sits silently in the Smithsonian Museum, continuing to inspire and awe as the machine that defeated the world champion [15]. Meanwhile, computer chess programs that play as competently as Deep Blue, and in some instances even better, can be downloaded

Figure 4: Deep Blue vs Garry Kasparov, New York City, 1997

Game Six between Garry Kasparov and the IBM supercomputer Deep Blue at the eighth move. Here, Deep Blue, playing White, delivered a crushing piece sacrifice to Kasparov who resigned only eleven moves later, losing the match decisively. What was Deep Blue's crushing move?

[6]In chess, a player gets no points for a defeat, one point for a victory, and one-half point for a draw. In this case, one loss, three wins, and two draws give a score of four for Kasparov.

from the internet and played on a modern laptop computer.

2 200 Million Evaluations Per Second: Understanding the Computer Grandmaster

> It was an impressive achievement, of course, and a *human* achievement by the members of the IBM team, but Deep Blue was only intelligent the way your programmable alarm clock is intelligent. Not that losing to a $10 million alarm clock made me feel any better.
>
> – Garry Kasparov[16]

Returning once again to the mathematical foundations of Deep Blue's success, we will now address two topics that were lightly touched upon in the preceding paragraphs. The first of these is the evaluation function itself, Shannon's $f(P)$: What is the evaluation function actually? How does it evaluate? Why is an evaluation function a skillful mechanism to enable computers to engage in the game of chess? The second topic is related to the first by the question, How is the evaluation function used algorithmically? Once a board is evaluated, how does a computer make an informed decision to play the best move? The answer, as the reader will discover, is more complex than, "The computer evaluates the best position that can arise from moving one of its pieces, and then makes the move that gives rise to that position."

It is important first to realize that the game of chess can be divided into three unique phases: the opening (where pieces develop and the opponents try to achieve an advantage in the position), the middlegame (where pieces are exchanged and the position is materially simplified), and the endgame (where the remaining forces rally to deliver mate or reach a drawn position). The computer relies on a different set of routines to respond to its opponents decisions in the distinct phases, so a treatment of the computer's *modus operandi* is given for each.

Opening play has long been dominated by chess theorists who have dictated the proper responses for both White and Black. *The Oxford Companion to Chess* recognizes 1,327 openings and variations in the modern repertoire of the theory [11]. While recall of such a massive repertory of openings may be a daunting challenge to man, it poses little difficulty to the computer. See Figure 5.

(a) English – Four Knights Var. **(b)** Ruy Lopez – Open Variation **(c)** Sicilian – Dragon Variation

Figure 5: Computers rely on extensive databases of openings to orchestrate their play in the beginning of the game. The opening is typically identified within the first or second turn of chess, at which time variations in the position are used to identify varying branches of play. These diagrams feature play through the fifth move, where the typical chess game lasts for thirty or forty moves.

Computers, if nothing else, are at least extremely capable of vast memorization, of storing the extensive treasuries of openings in memory, which can be accessed as required to specify the computer's best responses to opposing play. In fact, Shannon's evaluation function has little role in the first stage of chess, as the automaton's moves were predetermined by chess theorists decades, perhaps centuries, beforehand. In some sense, this is not the most elegant or attractive options for the computer to pursue, perhaps reducing the game to a kind of commit-it-to-memory savantism, yet it does invest the machine with a kind of "perfect memory," which is difficult to label as unfair "since that is the way chess masters play the opening" [30].

A portion of the endgame is handled similarly to the opening; it is played by brute memorization by so-called *endgame tablebases*. To understand what this is, it is useful at this point to define the concept of a *solved* game. When checkers was solved in 2007, researchers at the University of Alberta defined solving a game as "determining the final result [i.e. victory, defeat, or draw] in a game with no mistakes made by either player" [29]. Checkers was solved by an exhaustive examination of all possible positions that can arise in a game, which is 500 billion billion unique boards, in a careful analysis lasting eighteen years [34]. Chess, as a thirty-two piece board game, cannot be practically solved in this way by modern technology. This was shown to be the case first by Shannon in 1949, and has continued to be true even with modern computational advances in the present day. In particular, Shannon demonstrated that the minimum number of legal positions that can be obtained in chess is on the order of,

$$\frac{64!}{(32!)(8!)^2(2!)^6} \approx 10^{43} \tag{1}$$

a value that is *vastly* larger than 500 billion billion, and thoroughly impractical to store so many positions in memory[7].

Yet although chess in its initial position is presently unsolvable, it is possible to deduce the result of the game from immensely simplified boards. And in which phase of the game is the board typically at its least complex? In the endgame - and this is the concept of the endgame tablebase: to enumerate all checkmate positions that can be achieved with two Kings and some number of other pieces, and work backwards through the game to reveal how many moves must be played until mate is given from any given position [14]. Thus, the computer will be able to play perfectly, deliver checkmate in the optimal number of turns, or stave off defeat as long as possible once the position on the board is identical to a solution stored in the tablebase. These tablebases represent a partially solved version of chess, where the simplifications in the game allow for an exhaustive analysis of all possible variations. Tablebases currently exist for all positions with six pieces or less on the board.[8]

This reveals how the computer plays in the opening and in a portion of the endgame, but stills says very little of that phase of the game not dictated by theory or by solution databases. So enters the evaluation function.

2.1 Shannon's Evaluation Function: An Elegant Solution to Complex Play

Recall that Shannon based the idea of an evaluation function on several key attributes of chess that, by centuries of experience, were deemed important to the strength of a position. These key attributes were: piece values, King safety, Pawn structure, and piece development. Moveover, recall that every piece in chess is assigned a worth according to its strength relative to a Pawn. The King was infinitely valuable, the Knight was worth three Pawns, the Bishop slightly more than the Knight, the Rook five Pawns and the Queen nine. Then, in its most basic form, an evaluation function can be constructed as,

$$\begin{aligned} f(P) = \lambda(K - \dot{K}) + 9(Q - \dot{Q}) + 3(N - \dot{N}) + (3 + \epsilon)(B - \dot{B}) + \\ (P - \dot{P}) - \gamma(S - \dot{S}) + \delta(D - \dot{D})[30] \end{aligned} \tag{2}$$

where,

K,Q,B,N,R,P: The *number* of *K*ings, *Q*ueens, *B*ishops, K*n*ights, and *P*awns in the position belonging to the evaluating player. A King is considered to be lost (i.e. off the board) if he is in checkmate in the evaluated board.

S: A measure of the stability of the Pawn structure, captured perhaps by the sum of the number of isolated, doubled, or backward Pawns.

D: The development of pieces on the board, which is assigned a metric that asserts an equivalency between piece development and piece mobility. Mobility in this case is measured "as the number of legal moves available to" the evaluating player [30].

[7] In the interest of providing more recent news on the problems and theory of computer chess, Victor Allis in 1994 gave an estimate for the upper bound on the number of possible chess positions. In particular, he gave $(5)(10^{52})$ [2].

[8] It is possible to accurately evaluate any position with six pieces or less at the following website: Shredder Chess Six Piece Endgame Tablebase

ϵ**:** A small value greater than zero that is used to indicate the slight superiority of a Bishop over a Knight.

γ,δ**:** The coefficients on the worth of the Pawn structure and the piece development to positional strength. If these coefficients are large, then the computer will consider strong Pawn structure relative to the opposition and high piece mobility on the board as very desirable.

λ**:** The coefficient on the King term which, by necessity, must be relatively larger than the sum of all other coefficients. This is because the King must be kept in the position at any cost $-$ all other positional attributes being insignificant by comparison $-$ or else the game is lost. Shannon takes $\lambda = 200$.

In all cases let the dot superscript indicate that these are attributes of the position belonging to the opposition. Notice that a board position is evaluated as advantageous *only when* the evaluating player has some meaningful superiority in at least one of the given metrics, not when both positions are sound. Thus, the computer is not necessarily playing to constantly preserve sound Pawn structure, for example, but to force the opponent to accept a comparatively worse Pawn structure for himself.

While the evaluation function $f(P)$ as it stands is too high-level to be capable of playing a strategic game, it nonetheless represents the kernel of the idea that eventually gave rise to Deep Blue.

2.2 Minimax and Alpha-Beta Pruning

How is evaluation function used? Well, of course there are many such implementations in chess automata that incorporate an $f(P)$. Here is one for example from Shannon [30]: Let P be the present state of the board and let it be the computer's turn to move. The computer has available to it a range of legal moves that must be examined. In the typical position, this set of moves might include sensible moves like Pawn advances, Castling, and checks against the opponent's King. Of course, this set has numerous nonsensical moves included in its ranks (sacrificing the Queen into a random Pawn, for example). Yet even these moves are necessary to consider and evaluate since the computer has no way of intrinsically knowing what constitutes a strong positional move before an evaluation has occurred.

Of greater importance than the set of moves available to the computer player is the set of positions that result from them. Because, presumably, the computer is playing to win, it will play that move available to it such that Shannon's evaluation function $f(P)$ is maximized for the resulting board. Thus, the computer is playing its best move *according to the evaluation function.* Of course, some readers may have already guessed the impending disaster that comes inherent in this approach, which is that the computer evaluates only one move ahead in this case. It is all too often the case in chess, as in other strategic games, that what appears a winning attack quickly falls to shambles after some unforeseen consequence is revealed later on in the game.

A more sophisticated method of finding the best move is called *Minimax*, which determines the best move for the computer by also considering the consequences if the opposition should play their best move in response. At a search depth of two, the logic is as follows: For every move that the silicon player can make in the next turn, the opposition has a set of response moves among which there is a *best move.* What is the opponent's best move? It is the move such that when the resulting position is evaluated, $f(P)$ is a minimum. (The opponent plays to minimize the evaluation function because a board that evaluates badly is a good result for the opposition.) Thus the computer aims to maximize the evaluation function under the assumption that the best move is played in response. In such a case, the computer ensures that its position on the board is stronger than the positions resulting under any other variations of play calculated to a search depth of two. If m_i is the first move by the evaluating player, and $m_{i,j}$ is the opponent's response to that move, then mathematically this optimization problem is written out as,

$$\max_{m_i} \quad \min_{m_{i,j}} \quad f(P_{m_{i,j}}) \tag{3}$$

And if one is similarly inclined to search to a depth of four moves, then it is necessary to optimize the components of the following expression,

$$\max_{m_i} \quad \min_{m_{i,j}} \quad \max_{m_{i,j,k}} \quad \min_{m_{i,j,k,l}} \quad f(P_{m_{i,j,k,l}}) \tag{4}$$

These expressions are instructive for a second purpose, and that is to illustrate how the algorithm *minimax* arose and found its name (that is, from the lengthy string of min-max operators in each formulation). Indeed, the process just described is at the heart of all computer chess software, and is responsible for ensuring that the positional evaluations are handled in a sound and strategic way.

But one can, by inspection, quickly begin to see how the deeper depths get out of hand very quickly as the number of possible variations explodes dramatically. The average number of moves available to either player for a given board is 37 [17] and, therefore, the average number of variations the computer must consider at depth n is on the order of 37^n. Of course, some of these positions may be duplicated (by means of, for example, one board transposing into another), yet the issue remains: What can one do to tame the immense growth of variations in chess? The second algorithm at the heart of chess software is *alpha-beta pruning*, which, simply put, "shaves off" variations in play that are known to be superfluous, irrelevant to the computer's ability to skillfully evaluate a board.

Alpha-beta pruning is less intuitive than minimax and is perhaps best understood by simple example. Consider the all-too-common board position shown to the right in Figure 6. The Black King may legally move to only one of three available squares, these being g8 to his right, e8 to his left, and e7 to his lower left. All other moves would leave the King under attack and are therefore impermissible. If Black wishes to draw this game, how should he play? It is the case that moving the King to e7 loses immediately to Pawn to h7, threatening promotion to a Queen on the next term, and spelling a quick loss if White plays correctly. Alternatively, White may also move his King to g7 with the plan to bring the Pawn to the eighth rank, which is also winning, though less efficient than the immediate Pawn to h7 advance. Moving the King to e8 also fails to bring about a better outcome for identical reasons. The only remaining square for Black in this case is g8 and the hope that this will bring salvation.

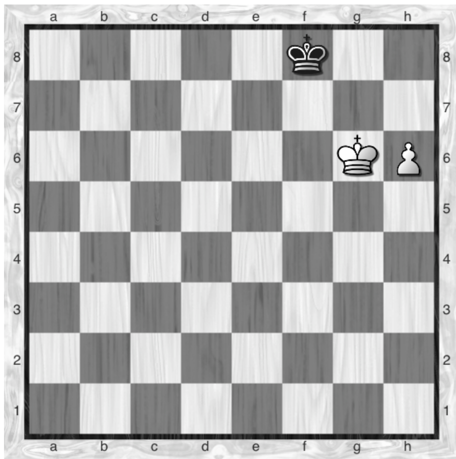
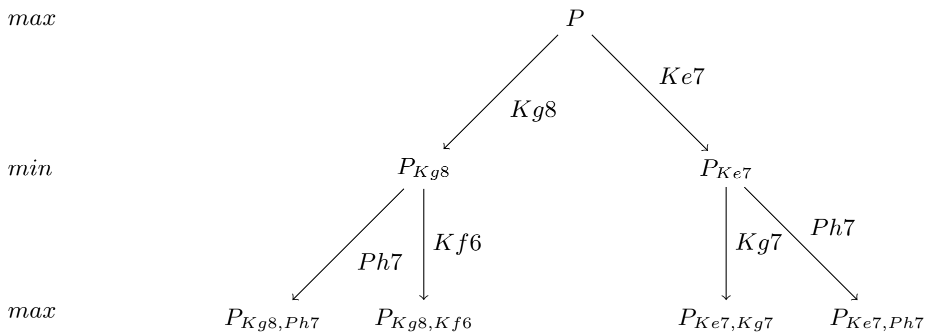


Figure 6: In this position, White is threatening to promote his Pawn to a Queen and, in doing so, dash all of Black's hopes for a draw. Indeed, since Black has lost all of his offensive pieces, the best he can hope for is a draw. If it is Black's turn to move, will he be able to stave off a defeat, and what role will alpha-beta pruning play?

Fortunately for Black, with King to g8, he has achieved a position that can be drawn, regardless of White's responses. The number of variations that lead to a draw are expansive, and it is beyond the scope of this paper to address them all, but hopefully one logical variation will be instructive of the theme. So, if the Black King is on g8, then one continuation is Pawn to h7, hoping to promote to Queen anyway. (Other variations, for example, King to f6 are possible, but all end in drawn games. Since this is a three piece endgame, in fact, it would be possible to set up the position in tablebase and confirm that correct play will not lead to defeat for Black.) In response, Black moves to h8, and White cannot help but draw the game in the next move. A drawn position here will be reached either by lack of sufficient material to deliver checkmate, or by failing to provide Black with any legal squares to which to move his King. Clearly then, Black does best to move his King to g8 in the position shown in Figure 6.

Suppose that a computer, playing Black, would also like to find the best move − the drawing move − King to g8, and that the computer is set up to search the tree of possible variations up to a depth of three. The set of circumstances confronting the computer could then be summarized in the following tree if P is the state of the board shown in Figure 6. Because initially moving the Black King to e7 or e8 lose in precisely the same way, the e8 case is excluded for concision.[9]

[9]Notice that this tree diagram is simplified from that used by a computer, which would contain all variations and be far larger and more complex. The goal was to create a digestable and illustrative tree diagram, rather than a perfectly accurate one.



The results of Shannon's evaluation function for this tree might be reported as follows (recall that drawn positions evaluate as zero, while losing positions evaluate negatively):

$$f(P_{Kg8,Ph7}) = 0 \qquad f(P_{Kg8,Kf6}) = 0 \qquad f(P_{Ke7,Kg7}) = -18 \qquad f(P_{Ke7,Ph7}) = -20?$$

Notice the question mark at $f(P_{Ke7,Ph7})$ – that will be addressed shortly. Here the "nodes" of the tree represent boards and the connecting edges represent the moves that transition from one position to another. If the minimax algorithm with alpha-beta pruning proceeds down chess tree to determine the best move for Black, it begins at the initial board position P and plays King to g8 to observe the results. From here, the computer has not exhausted its depth, so it continues by supposing that White plays Pawn to h7. After these moves, the specified search depth is reached and the computer evaluates the position and finds $f(P_{Kg8,Ph7}) = 0$. Similarly, if the opponent had played, instead of Pawn to h7, King to f6, the silicon player evaluates the resulting board and finds also that $f(P_{Kg8,Kf6}) = 0$. The assumption here is that the opponent would play the move that is worse for the evaluating player – but since both positions evaluated equally, either move will suffice and the value of $f(P_{Kg8})$ is zero as a result.

Because the evaluating player tries to maximize the evaluation function, it is known that the value of $f(P)$ is no less than zero, since a score of zero may be obtained by immediately playing King to g8. The alternative option considered by the computer from the initial board is King to e7, which was determined to be losing. The computer begins by imagining that it had played King to e7. In response, White plays Kg7 and the computer evaluates the position that emerges. There is no way to stop White from advancing the Pawn to the eighth rank, so the game is lost for Black if White plays intelligently. Evaluating $f(P_{Ke7,Kg7})$ is assumed to return negative eighteen, reflecting the totally hopeless situation for the computer.

Here, then, is the penultimate question: if the computer has already found that $f(P_{Ke7,Kg7}) = -18$, should the evaluating player bother to evaluate $f(P_{Ke7,Ph7})$? The intuitive reaction is probably "Yes" if only because it is another option available to the opposition, and the computer should be aware of all the variations. In actuality, however, the answer is "No." Recall that one of the fundamental assumptions of minimax is that the opponent strives to force the evaluating player to accept as little advantage as possible by responding to every move in the optimal fashion. Because the opponent tries to minimize Shannon's function, the value of $f(P_{Ke7})$ will be no greater than negative eighteen. But it is known that the computer can do no worse than zero by playing King to g8 from the start. Because the computer knows that it will definitely do worse by playing King to e7, it is a waste of time to continue evaluating variations within that branch of the tree.

The value of $f(P_{Ke7})$ is therefore not larger than zero, and the computer, wishing to maximize, will find the best move by playing to maximize its $f(P)$. Since the maximum of zero (given by playing King to g8) and negative eighteen (given by playing King to e7) is zero, the computer selects King to g8 as its desired move. This is the logic that defines the alpha-beta pruning algorithm, and allows for the computer to dismiss very large branches of the tree as irrelevant. In turn, this prevents the computer from being overwhelmed by the vast expanse of variations that must be accounted for in chess. While the minute details of the alpha-beta algorithm are beyond the scope of this paper, interested readers are directed to reference [39].

3 The Future of Computer Chess

> For every door the computers have closed they have opened a new one.
> − 2012 FIDE World Champion Viswanathan Anand[3]

With the defeat of Garry Kasparov in 1997, one might begin to wonder where the field of computer chess leads to next. Today the computer is an even stronger opponent than it was in 1997, but chess remains a complex and exciting sport between humans and computers alike. Indeed, just as there continues to be the FIDE World Championship, there is now too the equivalent World Computer Chess Championship, both of which continue to enthrall audiences annually. To adapt the words of English Grandmaster Nigel Short:

> Chess is a vast jungle − deep, relatively unexplored and slow to yield its myriad secrets − . . . It must retain an element of mystery [31].

The complete solution to chess remains, and probably shall continue to be, an open challenge to the computer and to computer scientists. Moreover, the curiosity that chess-playing automata spawned has produced questions of even greater interest. This is the essential truth that was captured by Viswanathan Anand in the quote that began this section: that computers, in their pursuit to, for instance, play chess, have offered much to science in return. Claude Shannon's claim that success in computer chess would serve as "a wedge in attacking other problems of a similar nature and of greater significance" [30] has also proven true in the motifs of machine learning, pattern recognition, and data mining technologies that have resulted in advancements in new drug therapies, more efficient marketing, and risk management. Indeed, the field opened up by chess machines is expansive and fertile ground for new exploration.

4 Answers to the Chess Exercises

> Today, chess programs have become so good that even grandmasters sometimes struggle to understand the logic behind some of their moves.
> – Grandmaster Kenneth Rogoff[28]

This section uses the *algebraic* methodology for recording and expressing chess moves. This is very standard in the chess community. For an introduction to algebraic notation in chess, refer to: Wikipedia: Algebraic notation (chess)

- Magnus Carlsen vs Helgi Gretarsson
    1. Bg6+ Rxg6 (1. ... Kxg6 2. Qh5#)
    2. Qe7#
- Deep Blue vs Garry Kasparov
    1. Nxe6

References

[1] *2001: A Space Odyssey*. Dir. Stanley Kubrick. Prod. Stanley Kubrick. By Stanley Kubrick, Arthur C. Clarke, Geoffrey Unsworth, and Ray Lovejoy. Perf. Keir Dullea, Gary Lockwood, and William Sylvester. Metro-Goldwyn-Mayer, 1968.

[2] Allis, Louis Victor. *Searching for Solutions in Games and Artificial Intelligence*. Thesis. Rijksuniversiteit Limburg Te Maastricht, 1994. N.p.: n.p., n.d. Print.

[3] Anand, Viswanathan. "Chess Is like Acting." Interview by Ansbert Kneip and Maik Groekathfer. n.d.: n. pag. *ChessBase.com*. Web. 21 Dec. 2012. http://www.chessbase.com/newsdetail.asp?newsid=4933.

[4] Aronian, Levon. "Aronian: 'I Have a Lot of Blood in My Brain' " Interview by Christian Schwager and Markus Lotter. *Berliner Zeitung* 20 Nov. 2008: n. pag. ChessBase.com. Web. 22 Dec. 2012. http://www.chessbase.com/newsdetail.asp?newsid=5025.

[5] Bellis, Mary. "History of Computer Chess - Dietrich Prinz Wrote the Chess Playing Program for a General Purpose Computer." *About.com Inventors*. About.com, n.d. Web. 10 Dec. 2012.

[6] Campbell, Murray, A. J. Hoane, Jr., and Feng-hsiung Hsu. *Deep Blue*. N.p., 1 Aug. 2001. Web. 24 Dec. 2012.

[7] Friedel, Frederic. "Garry Kasparov vs. Deep Blue." *ChessBase Columns*. N.p., May 1997. Web. 24 Dec. 2012.

[8] Greco, Gioachino, and Francis Beale. *The Royall Game of Chesse-play Sometimes the Recreation of the Late King, with Many of the Nobility : Illustrated VVith Almost an Hundred Gambetts*. London: Printed for Henry Herringman ..., 1656. Print.

[9] Greenblatt, Richard D., Donald E. Eastlake, and Stephen D. Crocker. *The Greenblatt Chess Program*. Massachussetts Institute of Technology, Aug. 1967. Web. 10 Dec. 2012.

[10] Gupta, Satyavarapu. "Singularity Free N-body Simulations Called 'Dynamic Universe Model' Don't Require Dark Matter." The Smithsonian/NASA Astrophysics Data System. Smithsonian Astrophysical Observatory, July 2010. Web. 25 Dec. 2012. http://adsabs.harvard.edu/abs/2010cosp...38.3683N.

[11] Hooper, David; Whyld, Kenneth (1992). *The Oxford Companion to Chess* (2 ed.). Oxford University Press.

[12] Horowitz, I. A. *How to Win in the Chess Openings*. New York: Simon & Schuster, 1985. Print.

[13] Hsu, Feng-hsiung. *Behind Deep Blue: Building the Computer That Defeated the World Chess Champion*. Princeton: Princeton UP, 2002. Print.

[14] Hurd, Joe. *Formal Verification of Chess Endgame Databases*. Oxford University, Aug. 2005. Web. 13 Dec. 2012.

[15] Kanellos, Michael. "Deep Blue Goes to Smithsonian – CNET News." *CNET News*. CBS Interactive, 24 Oct. 2002. Web. 24 Dec. 2012.

[16] Kasparov, Garry. "The Chess Master and the Computer." Editorial. n.d.: n. pag. *The New York Review of Books*. NYREV Inc., 11 Feb. 2010. Web. 21 Dec. 2012.

[17] Keene, Raymond D., and David N. L. Levy. *How to Beat Your Chess Computer*. London: Batsford, 1991. Print.

[18] Leopold, Todd. "The Meaning of Life." *CNN*. Cable News Network, 28 Apr. 2005. Web. 23 Dec. 2012.

[19] Levitt, Gerald M. *The Turk, Chess Automaton*. McFarland & Company, Inc., Publishers Jefferson, North Carolina and London. 2000.

[20] Levy, David N. L., and Monroe Newborn. *More Chess and Computers: The Microcomputer Revolution, the Challenge Match*. Potomac, MD: Computer Science, 1980. Print.

[21] Levy, David. "Computer ChessPast, Present and Future", Chess Life & Review, December 1973.

[22] McCrary, John. *Chess and Benjamin Franklin – His Pioneering Contributions*. Rep. BenFranklin300.org, n.d. Web. 8 Dec. 2012.

[23] McFarlat, Thomas. "History of Chess." *History of Chess*. N.p., 20 Nov. 2011. Web. 08 Dec. 2012.

[24] Meyer-Kahlen, Stefan. *Shredder Classic 4*. Computer software. *Shredder Computer Chess*. Vers. 4.12.0 (demo). N.p., n.d. Web. 7 Dec. 2012.

[25] Meyer-Kahlen, Stefan. "Welcome to Shredder Chess." *Shredder Computer Chess*. N.p., n.d. Web. 21 Dec. 2012.

[26] Poe, Edgar A. "Maelzel's Chess-Player." *Edgar Allan Poe Society of Baltimore*. The Southern Literary Messenger, Apr. 1836. Web. 08 Dec. 2012.

[27] Racknitz, Joseph. *Engraving of the Turk*. Digital image. *Mastering the Game: A History of Computer Chess*. Library Company of Philadelphia, 1789. Web. 8 Dec. 2012.

[28] Rogoff, Kenneth. "Grandmasters and Global Growth." Editorial. *Project Syndicate*. Project Syndicate, 5 Jan. 2010. Web. 21 Dec. 2012.

[29] Schaeffer, Jonathan, et. al. "Checkers Is Solved." Diss. University of Alberta, 2007. Abstract. *Science Magazine* 317.5844 (2007): 1518-522. Print.

[30] Shannon, Claude E. *Programming a Computer for Playing Chess*. Bell Telephone Laboratories, Inc., 8 Nov. 1949. Web. 9 Dec. 2012.

[31] Short, Nigel. "The Sunday Chess Column." Editorial. *The Telegraph* 29 Aug. 2004: n. pag. Web. 22 Dec. 2012.

[32] Silman, Jeremy. "Positional Ponderings." (2012): n. pag. *Chess.com Articles*. Web. 9 Dec. 2012. http://www.chess.com/article/view/positional-pondering.

[33] Silver, Alexandra. "Top 10 Man-vs.-Machine Moments." *Time Magazine*. Time, 15 Feb. 2011. Web. 24 Dec. 2012.

[34] University of Alberta. "Scientists Solve Checkers." *ScienceDaily*, 20 Jul. 2007. Web. 13 Dec. 2012.

[35] Wall, Bill. "The Slate/Atkin Program and CHESS X.x." *Chess.com*. Chess.com, 23 Sept. 2011. Web. 25 Dec. 2012.

[36] Weisstein, Eric W. "Chess." From MathWorld - A Wolfram Web Resource. http://mathworld.wolfram.com/Chess.html.

[37] Wright, Robert. "Can Machines Think?" *Time Magazine* 25 Mar. 1996: n. pag. Web. 24 Dec. 2012.

[38] "A History of Computer Chess from the 'Mechanical Turk' to 'Deep Blue'." *High Tech History*. N.p., 21 Apr. 2011. Web. 10 Dec. 2012.

[39] "Alpha-Beta Pruning." *Extra Credit Algorithm Walkthroughs*. N.p., n.d. Web. 26 Dec. 2012. http://www.cs.utah.edu/ hal/courses/2009S_AI/Walkthrough/AlphaBeta/.

[40] "Deep Bue: Overview." *Icons of Progress*. IBM, n.d. Web. 22 Dec. 2012.

[41] "Deep Bue: The Team." *Icons of Progress*. IBM, n.d. Web. 22 Dec. 2012.

[42] "Deep Blue - Kasparov, Game 1 of 1996 Match." *Chess.com*. Chess.com, n.d. Web. 23 Dec. 2012. http://www.chess.com/chessmentor/view_lesson?id=2841.

[43] "FIDE Top Players and Statistics." *World Chess Federation*. World Chess Federation, n.d. Web. 22 Dec. 2012.

[44] "GM Joel Benjamin." *USCF: The United States Chess Federation*. United States Chess Federation, n.d. Web. 23 Dec. 2012.

[45] "ICGA: ICGA Organisation." *ICGA: ICGA Organisation*. N.p., n.d. Web. 19 Dec. 2012.

[46] "Kasparov Beat Deep Blue." Deep Blue vs Kasparov (1996). Ed Schröder, n.d. Web. 24 Dec. 2012.

[47] "Mastering the Game: A History of Computer Chess." *Computer History Museum - Mastering the Game: A History of Computer Chess*. Computer History Museum, n.d. Web. 08 Dec. 2012.

[48] "The Pioneers of Computer Chess Programming." *Lewis Cup 2012/13 - Rainham Chess Club*. N.p., n.d. Web. 10 Dec. 2012.

[49] "A Computer Called Watson: Overview." *Icons of Progress*. IBM, n.d. Web. 22 Dec. 2012.

[50] "A Computer Called Watson: Transforming the World." *Icons of Progress*. IBM, n.d. Web. 22 Dec. 2012.