

Schach-Computer Programmiert auf Sieg

Uli Deker



Hagen Schmitt

Uli Deker: Schach-Computer programmiert auf Sieg

(Quelle: Bild der Wissenschaft – Dezember 1983) (photo copyright © by www.schaakcomputers.nl/) (600 dpi)

Schachcomputer im Taschenformat schlagen heute bereits die Hälfte aller Vereinsspieler. Wenn der Weg zur großmeisterlichen Spielstärke auch noch weit ist, wird doch klar: Computern gelang der Einbruch in eine Domäne der menschlichen Intelligenz. Wie spielen Computer Schach?

Informationswissenschaften

„Ich selbst gewinne nicht mehr gegen mein eigenes Programm“, bekennt Thomas Nitsche, ein junger Mathematiker, der zusammen mit seinem Kollegen Elmar Henne das Programm des einzigen kommerziellen bundesdeutschen Schachcomputers entwickelt. „Der Grund dafür ist, daß der Computer ähnlich wie der Mensch spielt, im Rahmen seiner Möglichkeiten aber keine Fehler macht. Sicher spielt ein Großmeister noch um einige Klassen besser. Ich glaube aber nicht, daß der Mensch dem Computer grundsätzlich überlegen bleiben wird.“

Sind Schachcomputer also intelligent? Das Schlagwort „künstliche Intelligenz“ läßt schnell die Emotionen hochgehen: Intelligenz als Charakteristikum menschlicher Lebewesen kann einer Maschine doch a priori gar nicht zukommen. Nicht zuletzt ist Intelligenz auch ein Faktor, der in unserer Gesellschaft eine große Rolle spielt und einhellig als positiv bewertet wird.

Andere wollen den Computer nicht zum stumpfsinnigen Rechenknecht herabgewürdigt sehen. Natürlich wird er erst durch die Intelligenz des Programmierers „intelligent“. Doch ist es nicht so einfach, daß der Computer genau das reproduziert, was der Programmierer in ihn hineingesteckt hat.

Auf dieser Ebene wird es aber stets ein Streit um Kaisers Bart bleiben. Interessant ist schließlich nur, was Computer tatsächlich können; in welchem Umfang sie Leistungen erbringen, die man beim Menschen der Intelligenz zuschreibt. Und: wie sie das machen.

Ein ideales Testgebiet hierfür ist das Schachspiel. Es vollzieht sich in einem überschaubaren Rahmen nach genau festgelegten Regeln. Niemand bestreitet, daß ein guter Schachspieler Intelligenz und sogar Intuition besitzen muß. Nicht umsonst gilt Schach als das „königliche Spiel“.

In der Tat simuliert es ja den Feldzug gegen ein feindliches Heer, das dem eigenen an Stärke völlig ebenbürtig ist. Allein durch taktische und strategische Überlegenheit kann es geschlagen werden. Umfangreiches Wissen, kombinatorische Vorausschau, aber auch List und Tücke machen den guten Schachspieler aus.

Die Vorreiter der künstlichen Intelligenz sahen hier von Anfang an die ideale Möglichkeit, ihre Thesen unter

Beweis zu stellen. Die Hoffnung, schon in den siebziger Jahren großmeisterlich aufspielende Computer präsentieren zu können, erfüllte sich allerdings nicht. Krasse Fehler unterliefen immer wieder, Ansätze zu längerfristigen Strategien waren kaum zu erkennen, im Endspiel wurden selbst Anfängeraufgaben wie das Matt-Setzen mit zwei Läufern nicht beherrscht.

Inzwischen gibt es Schach-Mikrocomputer für jedermann, die in der „ELO-Wertung“ bis zu 1900 Punkte erreichen. Sie spielen damit besser als 99% aller Schachspieler – Gelegenheitsspieler mitgerechnet – und immerhin besser als 50% aller Vereinsspieler. Von Groß- oder gar Weltmeistern trennen sie aber noch immer mehrere Klassen.

Wie spielen solche Computer Schach? Interessant ist ja weniger, daß sie 1900 ELO-Punkte erreichen, interessant ist das „Wie“.

Darum besuchte ich die Werkstatt des einzigen bundesdeutschen Herstellers und Entwicklers von Schachcomputern, Hegener + Glaser in München. Thomas Nitsche und Elmar Henne entwickeln die Programme, Ossi Weiner, dreifacher Münchener Schachmeister und ehemaliger Bundesligaspieler, unterstützt sie beim Testen. Gleichzeitig vertreibt er Schachcomputer.

Mathematisch gesehen ist Schach ein einfaches Problem, ein „endliches Zwei-Personen-Null-Summen-Spiel“. Im Klartext: Man weiß, daß es eine optimale Spielstrategie gibt. Unbekannt dagegen ist, wie diese Strategie aussieht – sonst wäre das Schachspiel „erledigt“ – und ob Weiß oder Schwarz bei dieser Strategie grundsätzlich unterliegen muß.

Grund für dieses Unwissen ist die Komplexität des Schachspiels. Man kann die Zahl der möglichen Stellungen einfach abschätzen, indem man überlegt, auf wieviele Arten die Figuren auf das Schachbrett gestellt werden können. Ganz grob betrachtet, hat man für die erste Figur, die man auf das Brett stellt, 64 Felder zur Auswahl. Berücksichtigt man, daß die Figur auch schon geschlagen sein kann, sind es sogar 65 Möglichkeiten. Für die nächste Figur bleiben noch 64, dann 63 Möglichkeiten, für die letzte schließlich 34. Das ergibt insgesamt $64 \cdot 63 \cdot \dots \cdot 34 \approx 10^{54}$ Möglichkeiten. Dagegen ist der

Uli Deker: Schach-Computer programmiert auf Sieg

(Quelle: Bild der Wissenschaft – Dezember 1983) (photo copyright © by www.schaakcomputers.nl/) (600 dpi)

Informationswissenschaften

Zauberwürfel mit seinen – noch dazu hochsymmetrischen – 10^{20} Möglichkeiten ein harmloses Kinderspielzeug.

Im Durchschnitt muß sich ein Schachspieler bei jedem Zug für eine von etwa 40 Möglichkeiten entscheiden. Im Prinzip könnte er das Spiel konsequent bis zu Ende durchdenken, indem er jede dieser 40 Möglichkeiten verfolgt, die jeweils wiederum durchschnittlich möglichen 40 Reaktionen seines Gegners untersucht, daraufhin wieder seine eigenen weiteren Zugmöglichkeiten, bis er schließlich auf eine erzwingbare Mattstellung stößt.

Diese Methode der rohen Gewalt scheidet natürlich an der Quantität: Nach n Zügen ergeben sich – grob abgeschätzt – 40^n mögliche Stellungen, eine mit n gigantisch anwachsende Zahl.

Computer können Stellungen sehr schnell berechnen. Dennoch reicht die durchschnittliche Bedenkzeit von drei Minuten pro Zug selbst Großrechnern

nur zur vollständigen Vorausberechnung von maximal sieben Zügen aus – also vier Zügen von Weiß und drei von Schwarz. Der größte Schachcomputer der Welt, „Belle“ von den Bell-Laboratories in den USA, erreicht damit eine Spielstärke von 2200 ELO-Punkten. Er rechnet 100 bis 1000mal schneller als die Mikrocomputer.

Rohe Gewalt ist aber nicht gerade ein Zeichen besonderer Intelligenz. Kein Schachspieler analysiert alle möglichen Züge, er konzentriert sich auf einige wenige, die ihm aufgrund seines Wissens und seiner Erfahrung besonders aussichtsreich erscheinen.

Hier setzt die Problematik des Schachprogrammierers ein. Grundsätzlich kann man die Situation bei jeder Stellung durch einen „Spielbaum“ darstellen: Die möglichen Züge führen als Äste weg vom Stamm des Baumes – der momentanen Stellung – und enden in neuen Stellungen. Von dort verzweigen sie sich weiter in die möglichen

Gegenzüge und so weiter. Die Verzweigungsstellen nennt man auch Knoten. Irgendwann wird von einem Knoten kein Ast mehr ausgehen, weil kein Zug mehr möglich ist: Eine Matt- oder Remis-Stellung ist erreicht.

Bei der Rohen-Gewalt-Methode wird dieser Spielbaum bis zu einer bestimmten Tiefe vollständig ermittelt. Belle beispielsweise berechnet pro Sekunde 200000 Knoten, um die Tiefe von maximal sieben Zügen zu erreichen. Die berechneten Stellungen müssen dann bewertet und miteinander verglichen werden – ein Problem für sich.

Ein „intelligentes“ Programm dagegen muß wie ein Schachspieler Züge selektieren, also jene Teile des Spielbaums abschneiden, die nicht den optimalen Zug zu enthalten scheinen.

Viele Zweige kann man von vornherein abschneiden, weil man sicher ist, bereits einen besseren Zug gefunden zu haben, oder weil für den Gegner

Minimax und Alpha-Beta

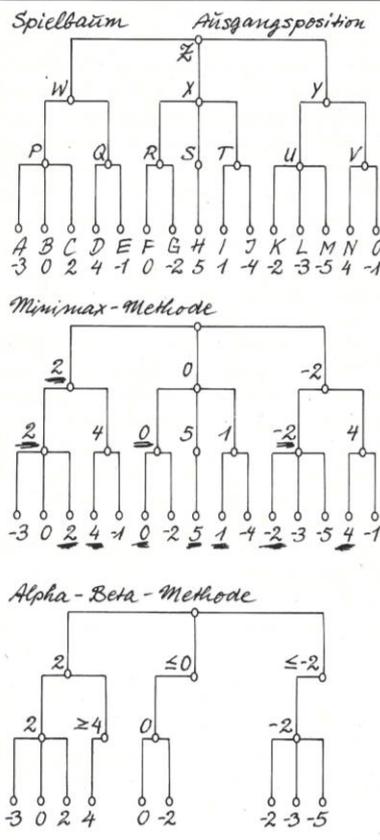
Um eine Schachposition vollständig zu analysieren, kann man ihren „Spielbaum“ aufstellen: Ausgehend von der Position trägt man zunächst alle den Regeln nach möglichen Züge als „Äste“ ein. Von jeder der entstehenden Positionen („Knoten“) trägt man wieder alle Gegenzüge ein. Dieses Verfahren wird so lange wiederholt, bis überall Positionen erreicht sind, von denen aus keine weiteren Züge mehr möglich sind: Matt- und Remisstellungen.

Natürlich scheidet dieses prinzipiell mögliche Verfahren an der Vielzahl der Züge. Auch ein Computer kann nur den Anfang des Baumes bis zu einer bestimmten „Tiefe“ aufstellen. Um den besten Zug auszusuchen, muß er alle erreichten Stellungen bewerten und muß zurückverfolgen, ob ihnen der Gegner nicht ausweichen kann.

Dazu dient die „Minimax-Methode“: Sie beruht auf der einfachen Überlegung, daß man selbst am Zug immer die Stellung mit maximaler Bewertung anstreben wird, der Gegner bei seinen Zügen aber die mit minimaler Bewertung (für einen selbst). Beim Zurückverfolgen zählt daher bei eigenen, von einem Knoten ausgehenden Zügen nur der beste, bei gegnerischen Zügen entsprechend der schlechteste.

Im Beispiel rechts wird Weiß von den Stellungen A, B und C die mit der höchsten Bewertung, also C, anstreben. Der Knoten P bekommt daher den Wert von C (2). Analog wird sich Weiß im Knoten Q für D entscheiden, also hat Q den Wert 4. Auf der nächsthöheren Ebene ist es umgekehrt: In W ist Schwarz am Zug und wird sich für die niedrigste Bewertung entscheiden, also für P. W übernimmt das Minimum der beiden Folgeknoten P und Q.

Meist ist es nicht notwendig, den Baum vollständig zu berechnen. Die „Alpha-Beta-Methode“ spart Rechenarbeit, ohne daß In-



formation verlorenght. Sie beruht darauf, daß auf einer „Minimum-Ebene“ immer nur der Knoten mit kleinstem Wert („Beta“), auf einer Maximum-Ebene der mit größtem Wert („Alpha“) zählt. Das Beispiel erläutert sie:

Für W zählt nur das Minimum von P und Q. Nach Berechnung von A, B und C steht $P = 2$ fest. Q interessiert nur dann noch, wenn sein Wert unter zwei liegt ($Beta = 2$). Aus $D = 4$ folgt aber bereits $Q \geq 4 > Beta$. Also braucht E gar nicht mehr untersucht zu werden. Spieltechnisch ausgedrückt: Schwarz wird sich von der Stellung W aus für P entscheiden, weil ihm sonst der Zug D droht. Welche weiteren Züge auf Q möglich sind, interessiert ihn nicht mehr.

Wirksam wird die Alpha-Beta-Methode aber nur dann, wenn gleich zu Beginn gute Züge gefunden werden. Beispielsweise muß D vor E untersucht werden.

Die „Killer-Heuristik“ geht dazu von folgender Annahme aus: Hat sich ein Zug bei einer Analyse als gut herausgestellt, so wird er mit großer Wahrscheinlichkeit bei der nächsten Analyse wieder als guter Zug abschneiden. Das Programm speichert daher solche Züge und beginnt die nächste Analyse mit ihnen. Bestätigt sich die Annahme, so kürzt die Alpha-Beta-Methode die Berechnung wesentlich ab. Im anderen Fall ist zwar nichts gewonnen, aber auch nichts verloren.

All diese Methoden führen zu einer exakten Analyse der Position bis zur vorgegebenen Tiefe. Sie können aber nicht über diesen „Horizont“ hinausblicken. Selbst ein Matt im nächsttieferen Zug entgeht ihnen. Ein „intelligentes“ Programm sollte daher mit flexibler Rechentiefe arbeiten. Es muß dann allerdings auf eine exakte Analyse verzichten. Es muß sich Teile des Spielbaums nach taktischen und strategischen Gesichtspunkten aussuchen – analog den menschlichen Schachspielern.

Uli Deker: Schach-Computer programmiert auf Sieg

(Quelle: Bild der Wissenschaft – Dezember 1983) (photo copyright © by www.schaakcomputers.nl/) (600 dpi)

eine gute Abwehrmöglichkeit besteht. Fehler des Gegners soll man schließlich nie voraussetzen. Für solch exaktes Abschneiden, das nachweisbar überflüssige Arbeit erspart, gibt es eine Reihe von Verfahren wie die „Alpha-Beta-Methode“ oder die „Killer-Heuristik“ (Kasten Seite 144). Sie sind aber nur technische Hilfsmittel, welche die Rohe-Gewalt-Methode effektiver ma-

chen, an ihrem Charakter aber nichts ändern.

Die Intelligenz beginnt beim eigentlichen Selektieren, beim Aussondern der Züge, die man aufgrund von Wissen und Intuition als aussichtsreich einstuft, ohne dies mathematisch exakt beweisen zu können.

Thomas Nitsche und Elmar Henne haben dazu ein Konzept entwickelt,

das sie „ π -Doppelstrategie“ nennen. Der Rechner untersucht dabei einen Ausschnitt aus dem Spielbaum, der nur wenige, dafür aber sehr lange Zweige enthält. In manchen Fällen denkt er bis zu 14 Züge voraus.

Die Suche nach einem guten Zug vollzieht sich in mehreren Schüben. Es kommt ja auch darauf an, Bedenkzeit für schwierige Situationen aufzusparen.

Die Analyse eines Zuges

Ein konkretes Beispiel zeigt am besten, wie ein Schachcomputer „denkt“. Der Ausgangspunkt ist eine Stellung (rechts) des sogenannten „Kopeck-Tests“, die als besonders schwierig gilt. Der beste Zug setzt nämlich zunächst ein Opfer von Weiß voraus. Das Auge eines menschlichen Schachspielers wird schnell erkennen, daß der weiße Springer auf dem Feld f6 beide schwarzen Türme gleichzeitig angreifen könnte. Allerdings deckt der schwarze Bauer g7 dieses Feld, er wird den Springer einfach schlagen. Dann aber kann der weiße Bauer e5 wiederum den schwarzen schlagen und in Richtung gegnerische Grundlinie marschieren. Der von Bauer und Läufer (a3) gleichzeitig bedrohte schwarze Springer muß wegziehen, der weiße Bauer zieht weiter auf f7 mit einem Gabelangriff auf beide Türme.

Um diese Zugkombination sicher zu erkennen, müßten Rohe-Gewalt-Programme wenigstens sieben Züge weit vorausrechnen, was allenfalls ein Großrechner wie Belle schafft.

Die π -Doppelstrategie von Thomas Nitsche und Elmar Henne nähert sich der Lösung trickreicher. In einer ersten Iteration (rechte Seite oben) werden nur direkte Züge verfolgt: Die Möglichkeit, mit dem Läufer den Springer zu schlagen oder die Dame anzugreifen; mit dem Turm den eigenen

Springer zu decken oder die g-Linie zu besetzen und den Bauern dort anzugreifen. Die Rechentiefe ist nicht starr festgelegt, sondern richtet sich nach der Dynamik der Stellung. Jede Endposition wird bewertet. Die nach dem Minimax-Prinzip gefundene momentan beste Zugfolge wird als „Hauptvariante“ gespeichert (1, 2, 3).

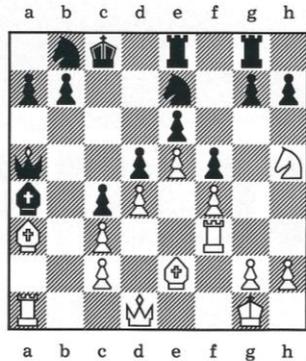
Die Hauptvariante ③ am Ende der ersten Iteration mit einer Wertung von 0,36 ist dem

Programm nicht gut genug. In einer zweiten Iteration betrachtet es nun insgesamt 24 Züge mit ihren Konsequenzen (unten). Nur acht offensichtlich schlechte Züge werden nicht verfolgt.

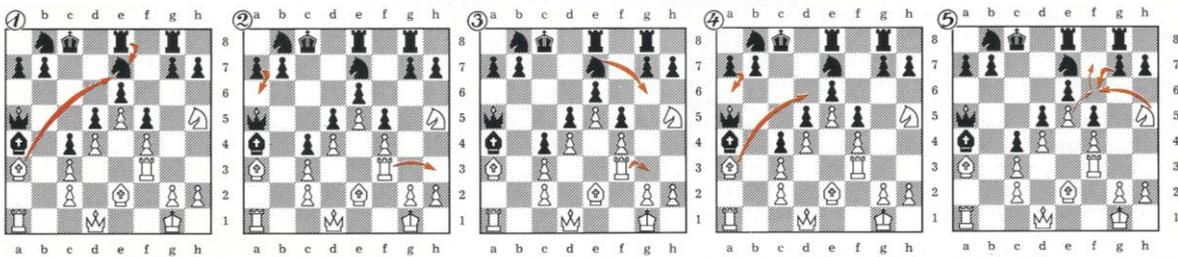
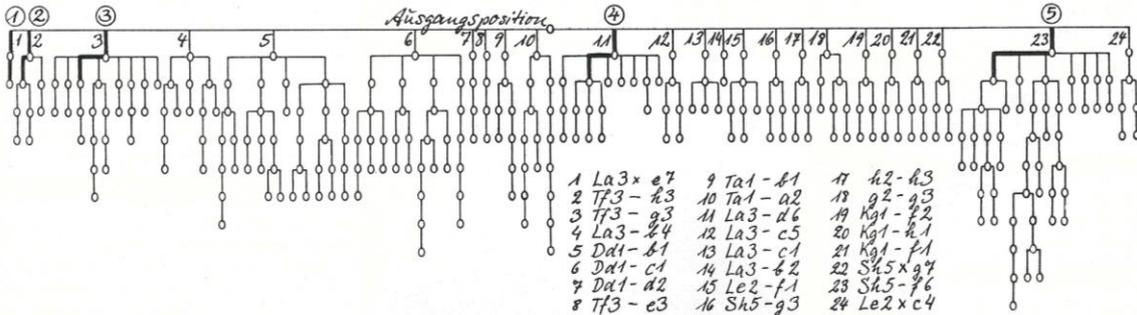
Schon rein optisch fällt auf, daß der Spielbaum relativ wenig Verzweigungen hat, dafür aber eine Tiefe von bis zu zehn Zügen: Nur interessant erscheinende Varianten werden untersucht, die aber so lange, bis eine ruhige Position erreicht ist. Dadurch kann das Programm den Wert des Opferzuges Sh5 – f6 entdecken, den es wegen des anfänglichen Materialverlustes erst an 23. Stelle analysiert (rechte Seite unten). Durch sinnvolle Selektion der Gegenzüge findet es bei allen Varianten den entscheidenden Gabelangriff des weißen Bauern (f6 – f7) oder bei Ausweichzügen von Schwarz den direkten Turmgewinn mit dem Springer.

Das Programm analysiert auch eine Verzweigungstat von Schwarz, den Angriff des schwarzen Läufers auf die weiße Dame. Es versucht zunächst, seinen eigenen Angriff fortzusetzen, findet dann aber den richtigen Gegenzug Dd1 x c2, der einen zusätzlichen Materialgewinn bringt und den eigentlichen Angriff nur aufschiebt. Daran scheitern Rohe-Gewalt-Programme.

Insgesamt berechnet das Programm 281 Knoten, um die endgültige Hauptvariante ⑤ zu finden. Es benötigt dazu auf einer PDP 11/34 dreieinhalb Minuten.



die Stellung Kopeck 7



Uli Deker: Schach-Computer programmiert auf Sieg

(Quelle: Bild der Wissenschaft – Dezember 1983) (photo copyright © by www.schaakcomputers.nl/) (600 dpi)

Auch ein Schachspieler wird nur kurze Zeit nachdenken, wenn er auf Antrieb einen offensichtlich guten Zug entdeckt.

Ein geradezu triviales Beispiel ist das Matt-Setzen. Besteht die Möglichkeit, in diesem oder dem nächsten Zug mattzusetzen, so wird kein Spieler lange nach Alternativen suchen. Ein anderes Beispiel ist das Schlagen der gegne-

rischen Dame. Dabei muß man aber schon überprüfen, ob der Preis für diesen Gewinn nicht zu hoch ist und er sich in den nächsten Zügen rächen wird.

Bei der „ersten Iteration“ wählt der Computer die Züge aus, die einen deutlichen Materialgewinn versprechen, und analysiert sie bis zu einer Tiefe von fünf Zügen.

Findet er dabei keine überzeugende Lösung, so untersucht er in einer „zweiten Iteration“ auch eigene Opfer. Erst in der dritten Iteration befaßt er sich mit reinen Stellungszügen, die nur die Position stärken.

Im Prinzip geht so auch ein menschlicher Schachspieler vor: Er verschafft sich zunächst einen Überblick über Möglichkeiten zum direkten eigenen oder gegnerischen Angriff. Erst dann wendet er sich „stillen Zügen“ zu. Kommentar von Ossi Weiner: „Erst das Normale, dann das Außergewöhnliche.“

Wichtig ist natürlich, auch taktische Motive zu erkennen. Ein typisches Beispiel ist der „Gabel-Angriff“: Ein Springer etwa bedroht zwei Figuren gleichzeitig und kann dadurch wenigstens eine von beiden schlagen.

Oder das „Abzugs-Schach“: Durch einen wegziehenden Läufer wird etwa eine Turmlinie geöffnet und dadurch der gegnerische König bedroht. Dieser Drohung muß zuerst begegnet werden, der Läufer kann inzwischen in Ruhe agieren.

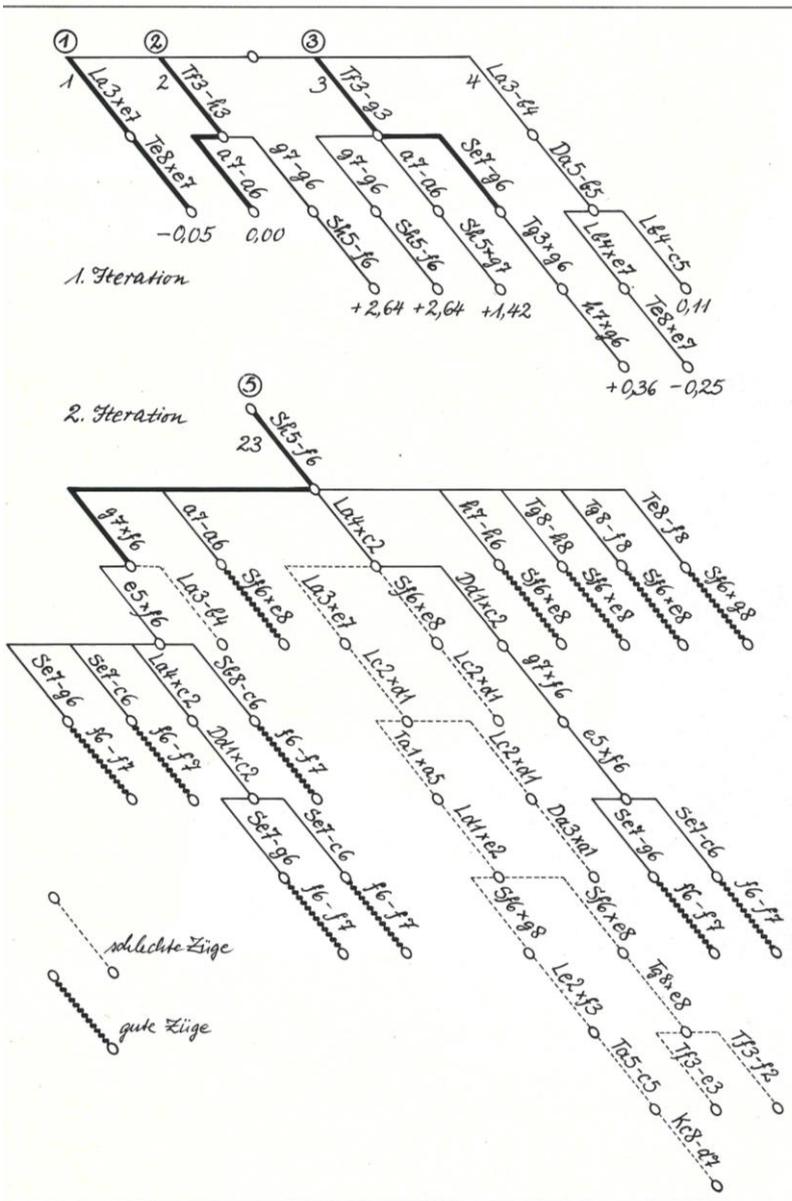
Ein weiteres Motiv, nach dem der Computer gezielt sucht, sind „überlastete“ Figuren, die beispielsweise zwei Figuren gleichzeitig decken müssen.

Diese Ausrichtung auf definierte Ziele ähnelt dem menschlichen Spiel wesentlich stärker als das blindwütige Durchrechnen aller Möglichkeiten. Die wenigen ausgewählten Züge können auch wesentlich weiter vorausberechnet werden. Als Grundsatz gilt: Abgebrochen wird erst, wenn eine stille Position erreicht ist. Instabile Situationen, in denen geschlagen werden kann, müssen weiter analysiert werden.

Beim positionellen Spiel untersucht der Computer bestimmte Muster. Zum Beispiel: Doppelbauern, zwei unmittelbar hintereinanderstehende Bauern, sind in aller Regel schädlich und werden vermieden. Dasselbe gilt für isolierte Bauern.

Umgekehrt wird ein guter Schachspieler immer ein Auge darauf haben, ob er einen Bauern auf die gegnerische Grundlinie bringen und damit in einen Offizier verwandeln kann. Auch haben sich bestimmte Bauern-Strukturen als effektiv für den Aufbau eines gut gedeckten Angriffs erwiesen.

Eine weitere positionelle Grundregel lautet: Besetze die Felder im Zen-



Uli Deker: Schach-Computer programmiert auf Sieg

(Quelle: Bild der Wissenschaft – Dezember 1983) (photo copyright © by www.schaakcomputers.nl/) (600 dpi)

Informationswissenschaften

trum des Bretts. Vierzig bis fünfzig solche Kriterien sind dem Computer als Ziele für die Selektion von Zügen vorgegeben.

Nachdem die Vorausberechnung der ausgesuchten Züge abgeschlossen ist, muß sich der Computer für eine der Varianten entscheiden.

Hauptkriterium dafür ist der zu erwartende Materialgewinn. Auch menschliche Schachspieler verwenden für die Bewertung der verschiedenen Figuren Standardzahlen. In der Regel

entsprechen Springer und Läufer drei Bauern, Turm fünf und Dame neun. Je nach individueller Spielweise verwenden Spieler aber auch davon abweichende Bewertungen. Manche ziehen die Springer den Läufern vor, andere legen mehr Wert auf die Läufer.

Auch der Computer kann Parameter wie diese variieren und durch Erfahrung und Ausprobieren optimieren. Neu entwickelte Programmvarianten spielen daher ständig gegen schon bestehende Programme und gegen Kon-

kurrenzmodelle. Und sie werden auf eine große Sammlung von Teststellungen „losgelassen“, um sich zu bewähren oder auszuschneiden.

Zieht man an dieser Stelle einen Vergleich zur Entwicklung des Schachspiels, so fallen die Ähnlichkeiten ins Auge: Die Regeln für „gutes“ Schachspiel sind allmählich gewachsen. Erst durch viele Spiele hat sich herauskristallisiert, was heute in den Lehrbüchern steht. In einer Art Evolution haben sich die guten Strategien nach und nach gegen die weniger guten durchgesetzt.

Der einzelne Schachspieler entwickelt sich wiederum ähnlich: Aus jeder Partie gewinnt er neue Erkenntnisse, zusätzlich lernt er Tricks und Methoden aus Lehrbüchern und durch das Studium von Meisterpartien.

In einer Partie entscheidet er sich für einen bestimmten Zug auf zweierlei Ebenen: Einerseits überprüft er die Position aufgrund seiner objektiven Kriterien, ein Prozeß, der im Bewußtsein abläuft. Daneben aber hat er ein „Gefühl“ für gute und schlechte Züge. Im Unterbewußtsein vorhandenes Wissen wird dabei mobilisiert.

Der holländische Psychologe de Groot hat in einer Studie festgestellt, daß sehr gute Schachspieler in den drei Minuten Bedenkzeit für einen Zug bis zu 180 Stellungen durchrechnen, also eine Stellung pro Sekunde. Diese vergleichen sie mehr oder weniger bewußt mit rund 50000 gespeicherten Einzelmustern. Die Übereinstimmung mit möglichst vielen Mustern bestimmt die Selektion der berechneten Züge und entscheidet letztlich auch die Auswahl des tatsächlich gemachten Zugs.

Damit zeigen sich die eigentlichen Probleme der Schachprogramme. Bezüglich Rechengeschwindigkeit können auch Mikrocomputer heute schon mithalten. Sie bewältigen etwa zwei bis drei Stellungen pro Sekunde.

Ein Vergleich mit 50000 Mustern aber ist zumindest heute noch utopisch – sowohl was die Speicherung als auch den Vergleich in so kurzer Zeit betrifft. Außerdem kennt ein Rechner kein Unterbewußtsein, der Programmierer muß alle Regeln und Erfahrungen ausdrücklich formulieren.

Auch das Lernen ist heute vorwiegend Sache des Programmierers und nicht des Computers selbst. Einmal



Schach-Weltmeister Anatolij Karpow trat in einer Simultanpartie gegen 18 Schachspieler und drei Schachcomputer an. Während die beiden Väter des Computerprogramms, Thomas Nitsche (oben links) und Elmar Henne (dane-

ben) gegen Karpow unterlagen, konnte einer der Computer wenigstens ein Remis erzielen (Partie Seite 154). Ebenfalls Remis gelang einem 14jährigen Schüler, alle anderen Partien gewann der Weltmeister souverän.

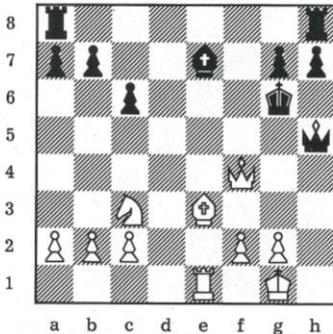
Uli Deker: Schach-Computer programmiert auf Sieg

(Quelle: Bild der Wissenschaft – Dezember 1983) (photo copyright © by www.schaakcomputers.nl/) (600 dpi)

Remis gegen den Weltmeister

Auch ein Weltmeister kann in einer Simultanvorstellung mit 21 Gegnern nicht alle Partien weltmeisterlich spielen. Dennoch zeigt das Remis gegen den Schachcomputer Mephisto, wie spielstark Mikrocomputer geworden sind.

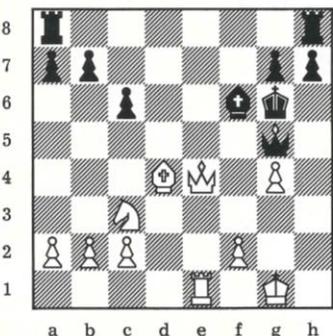
Karpow		Mephisto	
(1. September 1983)			
1. e2 - e4		e7 - e5	
2. Sg1 - f3		Sb8 - c6	
3. Lf1 - b5		Sg8 - f6	
4. 0 - 0		Sf6 x e4	
5. d2 - d4		e5 - e4	
6. Tf1 - e1		f7 - f5	
7. Sf3 x d4		Sf6 x Sd4	
8. Dd1 x Sd4		Ke8 - f7	
9. Lb5 - c4†		d7 - d5	
10. Lc4 x d5†		Le8 - e6	
11. Ld5 x e6†		Ke8 x Le6	
12. Te1 x Se4†		f5 x Te4	
13. Dd4 x e4†		Ke6 - f6	
14. Sb1 - c3		c7 - c6	
15. Lc1 - e3		Lf1 - e7	
16. Ta1 - e1		Dd8 - d7	
17. De4 - h4†		Kf6 - g6	
18. h2 - h4		Dd7 - f5	
19. h4 - h5†		Df5 - h5	



An dieser Stelle unterläuft dem Weltmeister ein kleiner Fehler. Durch g2 - g4 wäre ihm der Sieg sicher gewesen. Statt dessen zieht er:

20. Le3 - d4		Le7 - f6	
21. g2 - g4		Dh5 - g5	
22. Df4 - e4†		Kg6 - f7	
23. De4 - e6†		Kf7 - g6	
24. De6 - e4†		Kg6 - f7	
25. De4 - e6†		Kf7 - g6	
26. De6 - e4†			

Remis



programmiert, konserviert er sein Wissen, wird weder schlauer noch dümmer. Die Erfahrung mit selbstlernenden Programmen ist noch sehr gering und beschränkt sich beispielsweise auf die Optimierung von Parametern bei der Positionsbewertung.

Daß die Schwierigkeiten der Schachcomputer vor allem im Quantitativen liegen, zeigt ihre Schwäche im Endspiel. Wie man den König etwa mit Läufer und Springer mattsetzt, ist ein völlig berechenbares Problem, dessen Lösung in jedem Schachbuch steht. Dennoch scheitern hier die Schachcomputer, weil die Gesamtheit aller Endspielregeln so umfangreich ist, daß sie sich nicht in einem Mikrocomputer ab speichern läßt.

„Wir haben sehr schöne Erfolge beispielsweise beim Durchschieben von Bauern erzielt“, erklärt Thomas Nitsche. „Dennoch gibt es auch bei uns Endspielschwächen. Die einzelnen Regeln sind zwar recht einfach. Ihre Anzahl aber ist sehr groß.“

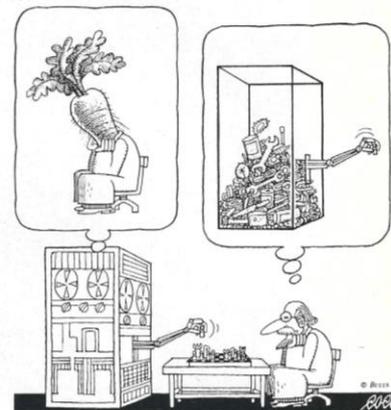
Etwas einfacher ist es, Eröffnungen abzuspeichern, da sie durch festgelegte Zugfolgen und nicht durch Regeln bestimmt sind. Thomas Nitsche: „Wir haben etwa 500 Varianten gespeichert, wobei wir auch Umstellungen in der Reihenfolge von Zügen erkennen und nicht gespeicherte Varianten, die aber zu denselben Stellungen führen.“

Der Speicherplatz für diese Eröffnungsbibliothek umfaßt etwa 5 K, das sind 5000 Zeichen oder zweieinhalb maschinengeschriebene DIN-A4-Seiten. Weitere 5 K sind nötig, um die Ein-Ausgabeprobleme zu lösen, also beispielsweise die Anzeige von Zügen oder die Annahme von Spieler-Anweisungen. Für die Berechnung des Spielbaums, vor allem auch die Spielregeln, sind 8 K nötig, 4 K für die Selektion von Zügen und weitere 8 K für die Bewertung der berechneten Stellungen. Das ist erstaunlich wenig.

Außerdem kann der Computer auch als Schachlehrer fungieren. Er weist den Spieler beispielsweise auf Fehler hin und führt ihm vor, welche Zugfolge ihm daraufhin droht. Auch kann die Spielstärke des Computers variiert werden. Er wählt dann ab und zu nur den zweit- oder drittbesten Zug aus. Wann, bestimmt ein Zufallsgenerator.

Auf die Frage nach der Zukunft antwortet Thomas Nitsche: „Wir werden

natürlich versuchen, noch mehr taktische Motive zu erkennen und die für die Selektion definierten Ziele um ein paar Züge weiter voranzutreiben. Auch das Endspiel muß noch verbessert werden. Ein Problem ist dabei natürlich auch die Hardware-Seite. Wir arbeiten im Moment mit einem Mikroprozessor, wie er auch in Personal-Computern verwendet wird. Mit einem speziell entwickelten Schachprozessor könnten wir die Geschwindigkeit und damit auch die Fähigkeiten des Pro-



gramms erheblich steigern. So arbeiten wir im Moment mit dem Prozessor 68000 von Motorola, der bereits vierbis achtmal schneller ist als der kommerziell eingesetzte.“

Sind Computer nun intelligent oder nicht? Meine Antwort lautet:

- Ohne die Intelligenz der Programm-Entwickler sind Computer natürlich Idioten.
- Durch die Künstliche-Intelligenz-Forschung werden aber typische Methoden des menschlichen Intellekts analysiert und für Computer zugänglich gemacht.
- Computer können dadurch Leistungen erbringen, die bisher nur mit Hilfe menschlicher Intelligenz möglich waren. Im beschränkten Rahmen ihres jeweiligen Einsatzes sind sie dem Menschen dabei sogar in punkto Fehlerfreiheit überlegen.
- An Komplexität und Flexibilität aber übertrifft das menschliche Hirn die Maschinen noch bei weitem. Lediglich spezielle Aufgaben in einem genau abgesteckten Rahmen werden Computern vorerst zugänglich sein – beispielsweise das Schachspielen. ♡

Uli Deker: Schach-Computer programmiert auf Sieg

(Quelle: Bild der Wissenschaft – Dezember 1983) (photo copyright © by www.schaakcomputers.nl/) (600 dpi)