

COMPUTERSCHACH

Die Hamburger Abiturenten Kai Himstedt und Wolfgang Stahn kamen mit einem selbst geschriebenen Schachprogramm unter die Landessieger im Wettbewerb „Jugend forscht“, Sparte Mathematik/Informatik. Sie nehmen nun damit am Bundeswettbewerb teil. Wir bringen nachstehend einen Auszug aus ihrer Preisarbeit.

Das Himstedt-Stahn-Programm

Angeregt durch die Computerschach-Welle und die kürzlich erzielten Erfolge des amerikanischen Schachprogramms CHESS 4.8 beschlossen wir, uns - im Rahmen des Wettbewerbs „Jugend forscht“ - mit dem Problem der Schachprogrammierung eingehender zu befassen.

Anfang des Jahres 1979 begannen wir mit der Entwicklung unseres Schachprogramms. Das Programm ist in der Programmiersprache FORTRAN IV geschrieben und hat zur Zeit eine Länge von rund 7.000 Zeilen. Es läuft auf dem Rechner Telefunken TR 440 am Rechenzentrum der Universität Hamburg. Das Programm arbeitet nach der Shannon B-Strategie. Es stellte sich aber bald heraus, daß das Programm auch mit gutem Erfolg nach der A-Strategie spielen kann. Einen Wechsel der Strategie erreicht man, indem man das Programm an jeder Verzweigungsstelle nicht nur mit wenigen plausiblen, sondern mit allen möglichen Fortsetzungen weiterrechnen läßt. Durch die aufwendige Vorbewertung, die ursprünglich zur Ermittlung der besten Fortsetzungen gedacht war, erreicht man durch die relativ gute Vorsortierung eine große Zahl an Alpha-Beta-Abschnidungen. Die maximale Verzweigungstiefe ist wegen des relativ langsamen Rechners derzeit auf 3 Halbzüge beschränkt. Es würde jedoch keine Schwierigkeiten bereiten, das Programm tiefer analysieren zu lassen.

Programmstruktur

Das Programm ist im wesentlichen aus zwei Teilen aufgebaut. Der eine Teil überprüft Spielerzüge auf ihre Richtigkeit und führt sie aus. Außerdem interpretiert er verschiedene Kommandos, mit denen sich beispielsweise die Verzweigungsparameter während des Spiels ändern lassen. Die Ausgabe des Schachbretts mit aktuellen Daten und des Computerzuges gehört ebenfalls zu diesem Teil. Der andere Teil übernimmt die Erzeugung von Zügen, die Baumsuche und die Bewertung von Stellungen.

Datenstruktur

Die Daten, die während des Spiels benötigt werden, sind größtenteils in Feldern mit 8 Zeilen und 8 Spalten abgelegt. Diese werden für jede Stellung im Spielbaum neu überarbeitet. Das Feld FIG enthält die Positionen der Figuren, entspricht also dem Spielbrett. Jedes Feldelement entspricht einem Feld auf dem Spielbrett. Die Figuren werden in Form von Ziffern dargestellt. Hierbei entspricht 1=Bauer, 2=Turm, 3=Springer, 4=Läufer, 5=Dame, 6=König. Die Computerfiguren erhalten ein positives, die Spielerfiguren ein negatives Vorzeichen.

Um die Zuverlässigkeit von Rochaden zu überprüfen, erhalten Könige und Türme, die noch nicht gezogen wurden, einen um 8 erhöhten Betrag. Der Wert 0 entspricht einem leeren Feld. Das Feld BEDVC enthält die Anzahl der Bedrohungen auf ein Feld des Spielbretts, die durch Computerfiguren hervorgerufen werden. BEDVS enthält entsprechende Bedrohungen durch Spielerfiguren. BEDGEG enthält für jedes Feld des Spielbretts die Anzahl von gegnerischen Figuren, die von diesem Feld aus bedroht werden, ZUGM die Anzahl der Zugmöglichkeiten der einzelnen Figuren. Steht auf einem bestimmten Feld des Spielbretts keine Figur, so sind die entsprechenden Feldelemente von BEDGEG und ZUGM ebenfalls 0. Zusätzlich sind in den Variablen ZUGMC bzw. ZUGMS die gesamte Anzahl der Zugmöglichkeiten für Computer bzw. Spieler gespeichert.

Baumsuche

Nachdem der Spieler einen Zug eingegeben hat und dieser ausgeführt wurde, hat das Programm die Aufgabe, selbst einen Zug zu

berechnen. Das Hauptprogramm ruft dazu das Unterprogramm COMZUG auf. Zuerst werden durch einen Aufruf des Unterprogramms ZUGFC alle legalen Zugmöglichkeiten des Computers erzeugt und in einem Feld gespeichert. Anschließend werden alle Züge der Reihe nach ausgeführt, vorberechnet (hierzu wird die gleiche Bewertungsfunktion wie für die Bewertung der Endstellungen verwendet) und zurückgesetzt. Zu jedem Zug wird die Vorbewertung gespeichert. Die so vorsortierten Züge werden danach ein zweites Mal ausgeführt (die besten Züge zuerst). Vor dem Zurücksetzen wird das Unterprogramm TIEFE2 aufgerufen. Dort werden zuerst wieder alle möglichen Gegenzüge erstellt und vorberechnet. Dann werden auch hier die Züge ein zweites Mal ausgeführt, und zwar die besten Gegenzüge zuerst. Vor dem Zurücksetzen wird von hier das Unterprogramm TIEFE3 aufgerufen. Diese Verschachtelung der Unterprogramme ließe sich theoretisch immer weiter fortsetzen. Bei unserem Programm ist jedoch TIEFE3 die Endebene bei der Verschachtelung.

Im Unterprogramm TIEFE3 werden zunächst alle Züge erstellt. Nun wird, falls möglich, der beste Zug vom vorhergehenden Aufruf von TIEFE3 ausgeführt (Killer-Heuristik). Anschließend werden alle Schlagzüge und Bauernumwandlungen und danach alle übrigen Züge ausgeführt. Das Programm merkt sich die beste Bewertung und gibt diese an TIEFE2 zurück. Das Programm prüft jedoch nach jeder Bewertung, ob es den beschriebenen Alpha-Beta-Algorithmus anwenden und so die Anzahl der zu bewertenden Stellungen stark reduzieren kann. In TIEFE2 merkt sich das Programm die kleinste Bewertung, die es aus TIEFE3 zurückgemeldet bekam, und gibt sie nach COMZUG zurück. Auch hier wird, wenn möglich, der Alpha-Beta-Algorithmus angewendet. Im Unterprogramm COMZUG merkt sich das Programm wieder wie in TIEFE3 die beste Bewertung und zusätzlich noch den dazugehörigen Zug. Diesen Zug führt das Programm dann als seinen endgültigen Zug aus und gibt ihn zusammen mit dem aktuellen Spielfeld auf der Konsole aus.

Über Parameter läßt sich variieren, mit den wieviel besten Zügen jeweils zur folgenden Tiefe verzweigt werden soll. Die Parameter können so gewählt werden, daß das Programm in jeder Tiefe mit allen Zügen weiterrechnet, also nach der Shannon A-Strategie spielt. Ebenso wird es dadurch möglich, die Maximaltiefe auf ein oder zwei Halbzüge zu begrenzen. Es ist auch möglich, über einen weiteren Parameter die Gesamtzahl der zu analysierenden Stellungen auf einen bestimmten Wert zu beschränken, wodurch eine sehr gute Zeitkontrolle ermöglicht wird. Hierbei wird nach jedem Aufruf von TIEFE2 überprüft, ob die eingestellte Maximalzahl erreicht oder bereits überschritten wurde. Ist dies der Fall, so analysiert das Programm keine weiteren Züge mehr, sondern führt den bis dahin besten Zug aus. Die Analyse wird ebenfalls sofort abgebrochen, wenn das Programm sieht, daß es mattsetzen kann. Hat der Computer nur eine Zugmöglichkeit, verschachtelt das Programm nicht, sondern führt den Zug sofort aus.

Bewertung

Die Bewertung von Stellungen im Spielbaum übernehmen zwei Unterprogramme: BEWFC und BEWFS. Das erste bewertet Stellungen, die nach Zügen der Computerfiguren entstanden sind. Das andere Unterprogramm bewertet Stellungen, die nach Zügen des Gegners (der Spielerfiguren) im Spielbaum auftreten. Wie weiter unten erläutert wird, sind zwei getrennte Unterprogramme notwendig, um Schlagwechsel und Bedrohungen richtig bewerten zu können. Die Bewertungsfunktionen werden sowohl für die Vorbewertungen als auch für die Bewertung von Endstellungen verwendet.

Vor jedem Aufruf eines der beiden Unterprogramme (d. h. bei der Bewertung einer neuen Stellung) erfolgt ein Aufruf des Unterprogramms AENDER. Dadurch werden die Felder und Variablen, die die Grundlage für die Bewertung bilden, überarbeitet und auf den aktuellen Stand gebracht.

Die Bewertungsfunktionen umfassen taktische und strategische Aspekte. Die Bewertungskriterien (Heuristiken) bauen auf allgemeinen Erfahrungsregeln im Schachspiel auf. Das wichtigste Kriterium ist die Bewertung des Materials und der Schlagwechsel und Bedrohungen. Für das Material gilt: Bauer=300, Springer=975, Läufer=1050, Turm=1500 und Dame=2700 Punkte. Vom Material her identische Stellungen werden durch die Positionsbewertung der übrigen Heuristiken (z.B. Entwicklung der leichten Figuren, Beherrschung der Brettmitte, Gesamtzahl der Zug-

Das Himstedt-Stahn-Programm (Kai Himstedt und Wolfgang Stahn)

(Quelle: <https://rochadeuropa.com/> Nr. 191 – Juni 1980) (photo copyright © by <http://www.schaakcomputers.nl/>) (600 dpi)

möglichkeiten etc) unterschieden. Die Positionsbewertung macht maximal etwa den Wert eines Bauern aus. Eine vollständige Auflistung aller Bewertungskriterien unseres Programms befindet sich im Anhang.

Das größte Problem bei der Bewertung von Stellungen ist die Analyse von Schlagwechseln. Die meisten Programme, darunter auch CHESS 4.8, verschachteln mit allen Schlagzügen solange weiter, bis die am Zug befindliche Seite keine Figuren des Gegners mehr bedroht. Nach Slate und Atkin, den Autoren von CHESS 4.8, verbraucht diese zusätzliche Verschachtelung ca. 20-70%, bei manchen Stellungen bis zu 95% der Gesamtrechenzeit (1).

Die Analyse von Schlagwechseln in der Bewertungsfunktion entfällt bei diesem Verfahren. Das Verhältnis von Aufwand zu Nutzen erschien uns jedoch sehr ungünstig. Deshalb entschlossen wir uns, Schlagwechsel in den Bewertungsfunktionen teils durchzurechnen, teils „durchzuschätzen“. Das Programm verkalkuliert sich dabei äußerst selten. Wie schon erwähnt, verwendet unser Programm zwei getrennte Bewertungsfunktionen. Dies ist aus dem einfachen Grund notwendig, da es bei der Bewertung von Bedrohungen und Schlagwechseln ein großer Unterschied ist, welche Seite am Zug ist. Ein Beispiel: Bei einer zu bewertenden Stellung im Spielbaum wäre der Gegner des Computers am Zug. Der Gegner bedroht einen nicht gedeckten Bauern des Computers. Der Computer muß damit rechnen, daß der Bauer beim nächsten Zug geschlagen wird. Aus diesem Grund bekommt er für diesen bedrohten Bauern 300 Minuspunkte. Würde der Computer in der gleichen Stellung einen Bauern des Gegners bedrohen, so würde er hierfür nur sehr wenig oder gar keine Punkte (abhängig von den Zugmöglichkeiten dieses Bauern) erhalten, weil der Gegner am Zug wäre und der Computer damit rechnen muß, daß sich der Gegner mit seinem nächsten Zug der Bedrohung entzieht oder den Bauern deckt. Ähnliches gilt für mehrfach bedrohte und gedeckte Figuren (Schlagwechsel).

Ein weiterer Vorteil bei unserer Methode ist, daß wir Matt und Patt in den Bewertungsunterprogrammen abfragen können, da uns die Felder für die Bedrohung und Zugmöglichkeiten der einzelnen Figuren zur Verfügung stehen. Im Gegensatz dazu „sieht“ CHESS 4.8 ein Matt erst dadurch, daß es beim Versuch weiterzuschachteln für die eigentlich am Zug befindliche Seite keine legalen Zugmöglichkeiten findet (1). Dies bedeutet beispielsweise, daß CHESS 4.8 4 Halbzüge tief verschachteln muß, um ein zweizügiges Matt mit Sicherheit zu erkennen, während unser Programm mit 3 Halbzügen auskommt. Auf der Rechenanlage, auf der unser Programm zur Zeit läuft, bewertet es 10-30 Stellungen pro Sekunde. Bei einer Verschachtelungstiefe von 2 Halbzügen sind es insgesamt 2.500 - 3.000 Stellungen.

Während der ersten Züge versucht das Programm, die jeweilige Ausgangsstellung in der Eröffnungsbibliothek zu finden. Ist die Stellung gespeichert, so wird der dazu vermerkte Zug sofort ausgeführt. Die Eröffnungsbibliothek enthält derzeit 215 verschiedene Stellungen aus 20 verschiedenen Schacheröffnungen.

Dialogmöglichkeiten

Anstelle der Eingabe eines Zuges kann der Benutzer eine Reihe von Kommandos geben, die vom Unterprogramm INPUT interpretiert und ausgeführt werden. Ein Kommando wird erkannt, wenn es vollständig oder in einer eindeutig abgekürzten Form eingegeben wird. Zur Zeit stehen ca. 25 verschiedene Kommandos zur Verfügung. Hierzu einige Beispiele: INVERS bewirkt, daß das Spielfeld umgedreht wird und daß das Programm mit den Figuren des Gegners weiterspielt. NULL bewirkt, daß der Spielerzug ausgelassen wird und das Programm sofort seinen nächsten Zug ausführt. PARAMETER ermöglicht eine Änderung der Verschachtelungsparameter. TRACE EIN bewirkt, daß nach jedem Aufruf der Bewertungsfunktionen das Spielfeld mit der aktuellen Bewertung ausgegeben wird. Dies ermöglicht eine gute Kontrolle der Vorgänge bei der Baumsuche. Beim Start des Programms wird das Spielfeld mit der Normalaufstellung der Figuren initialisiert. Zum Lösen von Schachproblemen etc. läßt sich jedoch jede beliebige Stellung aufsetzen.

Ausblick

Die Entwicklung unseres Schachprogramms ist noch lange nicht abgeschlossen. Die vordringlichste Aufgabe ist es, die Geschwindigkeit des Programms zu erhöhen. Man könnte z. B. häufig gebrauchte Unterprogramme in eine maschinennahe Programmiersprache umschreiben. Außerdem sind Verbesserungen bei der Baumsuche und bei den Bewertungsfunktionen vorgesehen. Eine Weiterverfolgung von Schachgeboten würde die Spielstärke des Programms wahrscheinlich ebenfalls beträchtlich erhöhen, da viele Angriffe mit Schachgeboten eingeleitet werden.

Kai Himstedt
Stephanstraße 101e
2000 Hamburg 70

Wolfgang Stahn
Im Winkel 8
2000 Hamburg 20

Literatur

1. Frey, Peter W. (Hrsg.)
Chess Skill In Man And Machine
Springer-Verlag, New York, 1977
2. Bell, Alex G.
The Machine Plays Chess?
Pergamon Press, Oxford, 1978
3. Bruderer, Herbert E.
Nichtnumerische Informationsverarbeitung, Verlag Linguistik,
9400 Rorschach/SG, 1979
4. Frey, Peter W. und Atkin, Larry R.
Creating a Chess Player
BYTE, Ausgaben Okt. bis Dez. 1971

Anhang

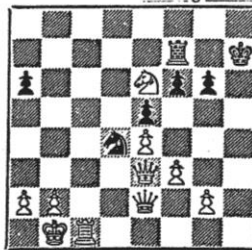
Bewertungskriterien des Himstedt-Stahn-Programms

Allgemeine Bewertungskriterien	Springer
Materialbilanz	Entfernung zum Zentrum
Zugmöglichkeiten	Deckung
Matt und Patt	Entwicklung
Bedrohungen	Entfernung zum feindlichen
bedrohte Felder	König
bedrohte Figuren (Schlagwechsel)	Läufer
Bauern- und Springergabeln	Abstand zum Zentrum
sonstige Gabeln gegen ungedeckte Figuren	Deckung
Schachgebote	Damen
Spezielle Bewertungskriterien	Entfernung zum Zentrum
Bauern	Deckung
Vordringtiefe	Dame nicht während der ersten Züge entwickeln
Entfernung zum Zentrum	Entfernung zum feindlichen König
Deckung	Könige
Mehrfachbauern	Entfernung zum Rand u. zu den Ecken (nicht Endspiel)
isolierte Bauern	Entfernung zum Zentrum (Endsp.)
Freibauern	Zugmöglichkeiten (nicht Endspiel)
Vordringtiefe	mittl. Abstand zu den Bauern (Endspiel)
Blockaden	Abstand der Könige voneinander (Endspiel)
Bauern in der 7. Reihe	Oppositionen d. Könige (bei elementaren Mattführungen)
Türme	Bauern in der Königsumgebung (nicht Endspiel)
Entfernung zum Zentrum	Doppelbedrohung i.d. Königsumgebung
Deckung	
Verdoppelung (nicht i. Endspiel)	
Besetzung offener Linien	
Entfernung zum feindlichen König	
Türme in der 7. Reihe (nicht im Endspiel)	

Partiebeispiel

Weiß: Himstedt-Stahn-Programm (Rechenzeit 15 Sek. pro Zug)
Schwarz: CHESS CHALLENGER 7 (Stufe 2, etwa 15 Sek. p. Zug)
1. e2-e4 e7-e5 2. Sg1 f3 Sb8 c6 3. Lf1-c4 Lf8-c5 4. d2-d3 Sg8-f6
5. Sb1-c3 d7-d6 6. 0-0 Lc8-e6 7. Lc4xe6 f7xe6 8. g2-g3 0-0 9.
Lc1-g5 d6-d5 10. Dd1-e2 Dd8-d6 11. e4xd5 exd5 12. Sc3-b5
Dd6-e7 13. Sb5-c3 Ta8-d8 14. Tf1-e1 Tf8-e8 15. De2-d2 De7-e6
16. Sc3-b5 Lc5-b6 17. Lg5-f4 a7-a6 18. Sb5-c3 d5-d4 19. Sc3-a4
Lb6-a5 20. Sa4-c5 La5xd2 21. Sc5xe6 Te8xe6 22. Lf4xd2 Sf6-g4
23. h2-h3 Te6-f6 24. Ld2-g5 Tf6xf3 25. Lg5xd8
Sg4xf2 26. Ld8xc7 Sc6-b4 27. Kg1-g2 Tf3-f7 28. Te1-e2 Tf7xc7
29. Te2xf2 b7-b5 30. Ta1-e1 Tc7-c5 31. Kg2-g1 Sb4xc2 32.
Te1-c1 e5-e4 33. Tc1xc2 Tc5xc2 34. Tf2xc2 e4xd3 35. Tc2-d2
Schwarz gibt „auf Rat seines Besitzers“ auf.

Lösung eines Mattproblems



Partie: Spassky - Kortschnoi,
Kiew 1968
Matt in 3 Zügen mit Damenopfer
1. Dh6+!
Benötigte Rechenzeit: 86 Sek.
Bewertete Stellungen: 1655
Das Programm macht diesen
Zug aufgrund des Bauernge-
winns auf g6.

Das Himstedt-Stahn-Programm (Kai Himstedt und Wolfgang Stahn)

(Quelle: <https://rochadeuropa.com/> Nr. 191 – Juni 1980) (photo copyright © by <http://www.schaakcomputers.nl/>) (600 dpi)

Internet: <https://chessprogramming.wikispaces.com/Kai+Himstedt>

<https://www.jugend-forscht.de/projekt Datenbank/die-entwicklung-eines-schachprogramms-als-beispiel-fuer-eine-spezielle-form-der-kuenstlichen-intelligenz.html>