Drosophilas Glück und Ende

Chrilly Donninger bot den folgenden Artikel einem österreichischen KI-Journal an, welches ihn ablehnte: kein Wunder, denn mit derselben Erfolgs-Chance hätte Chrilly den Zeugen Jehovas für den Wachtturm eine Schrift über Evolutions-Biologie anbieten können. CSS Online bringt den Artikel natürlich, weil Computer-Schächer schon lange aufgehört haben (sollten), an Künstliche Intelligenz zu glauben ...

Gott sah alles an, was er gemacht hatte: Es war sehr gut. Es wurde Abend und es wurde Morgen: der sechste Tag. Gen. 1,31

Wenn ich eine größere finanzielle Transaktion vorhabe, frage ich einen befreundeten Ökonomen über die vorherrschende Ansicht zu diesem Thema. Dann weiß ich mit einiger Sicherheit, wie es nicht wird, und richte mich danach. In meiner Profession ist die Sache noch einfacher. Die Prognosen John McCarthys und seiner AI-Jünger bilden einen Entropiesatz – den Raum der hoffnungslosen Lösungsansätze. Stanley Kubricks HAL und meine Windows-PCs haben nur eins gemeinsam: Manchmal geht gar nichts mehr.

1989 veröffentlichten J. Schaeffer und M.Donskoy "Perspectives on Falling from Grace". Die Confessiones zweier erfolgreicher akademischer Schachprogrammierer. Man habe das heilige Ziel der Computer-Drosophila durch die Gier nach dem Turniersieg verraten. Schaeffers Bekehrungserlebnis war eine bittere Niederlage gegen Chiptest. Er ist aber mit dem Dameprogramm Chinnook wieder rückfällig geworden.

Im selben Jahr arbeitete ich im European Space Technology Centre in Noordwijk/NL. In den feuchtkalten Winterabenden fühlte ich mich einsam und hatte Heimweh. Um dieses Gefühl zu betäuben, kaufte ich mir den Mephisto-Polgar-Brettcomputer von Ed Schroeder. Das Spielen gegen den Polgar half auch nichts, und so fasste ich eines Abends den Entschluss: Ich kann ein bisserl Schachspielen, ich kann programmieren, warum mache ich kein Schachprogramm? Das Ziel war ein PC-Programm, das es mit dem Polgar aufnehmen kann. Ich hatte damit die Droge hinreichend erhöht, das bisher deprimierende Trommeln des Regens wurde zur angenehmen Hintergrundmusik. Allerdings leide ich bis heute unter schweren Suchterscheinungen. Von McCarthys Drosophila hatte ich nicht den geringsten Tau, McCarthy war mir nur als Schöpfer einer eigenartigen Programmiersprache ein vager Begriff. Es wäre mir auch vollkommen egal gewesen: Mein Ziel war, das Heimweh zu betäuben und den Polgar zu schlagen. Es war eins der sieben fetten Jahren des Computerschachs. Ed Schroeder zehrt noch heute von den Polgar-Tantiemen.

McCarthys Thesen

Ebenfalls 1989 besiegte Chiptest die ersten Großmeister. Der erste Artikel über dieses Programm trug den programmatischen Titel „Designing a Single Chip Grandmaster while knowing nothing about chess“. Chiptest erlangte als Deep Blue Weltruhm. John McCarthy, der den Sieg über den Schachweltmeister einst als Aufwärmübung für die eigentlichen Aufgaben der AI formuliert hatte, reagierte in einer Buchbesprechung in *Science* auf Deep Blues Gipfelsieg säuerlich.

Three features of human chess play are required by computer programs when they face harder problems than chess. Two of them were used by early chess programs but were abandoned in substituting computer power for thought.

1) Human chess players cannot examine all moves at every position they think about and therefore must forward prune the move tree and select the more promising moves for exploration; early chess programs also pruned. About 1969 forward pruning was eliminated, and computer power was relied upon to examine all moves. It made the programs work better, because early programs sometimes pruned good moves. Eliminating pruning was possible, because there are only about 40 moves in a position. In the game of Go, there are up to 361 moves in a position, so even computers must forward prune.

McCarthy hat offensichtlich 30 Jahre Entwicklung ignoriert. Moderne Programme suchen nicht brute-force, sondern benützen dynamische Information, die Struktur des Suchbaumes, um Varianten sowohl zu erweitern als auch zu verkürzen. Mein Schachprogramm Hydra sucht in einer Mittelspiel-Stellung 17-18 Halbzüge tief. Die kürzesten Varianten haben eine Länge von 7, die längsten bis zu 40 Plies. Im Gegensatz zum Lochkarten-Zeitalter verwendet man heute „soft pruning“. Anstatt einen Zug gänzlich abzuschneiden, wird der Suchbaum verkürzt. Ein um 2 Plies reduzierter Suchbaum ist nicht viel teurer, als den Zug überhaupt wegzulassen. Das Programm hat aber die Chance, fehlerhaftes Pruning in den folgenden Iterationen zu erkennen, während die von McCarthy propagierte Methode selbst bei unendlicher Suchtiefe den Fehler nicht mehr korrigiert. Außerdem erzeugen diese verkürzten Suchen weitere Informationen über den Suchbaum, die in Folge profitabel verwendet werden können. Die meisten Programme berechnen nur die ersten 3 Züge „brute-force“, die restlichen Züge werden stark geprunt. Der optimale exponentielle Faktor beträgt bei Alpha-Beta 6, tatsächlich haben die Programme einen Faktor von 3-4. Ohne Sucherweiterungen würde der Faktor unter 3 sinken. Für Fritz‘ Programmierer Frans Morsch liegt daher die Intelligenz eines Programmes im Suchalgorithmus. In erster Näherung bestimmt die Bewertungsungsfunktion den Spielstil, die Suche die Spielstärke. Wer die besseren Suchheuristiken entwickelt, hat im Wettlauf der Spitzenprogramme die Nase vorn.

2) An early Soviet program could consider certain moves as analogous to moves that had been found to be bad in a parallel position. It would prune them unless something was observed that would rehabilitate them. This also was abandoned. Present programs spend most of their time rejecting the same moves millions of times apiece.

Wie ich aus persönlicher Kommunikation mit Feng Hsu weiß, hatte Deep Blue die vom Kaissa-Team entwickelte Methode der analogous moves implementiert. Laut Feng Hsu reduziert diese Methode den Suchbaum bis zum Faktor 10. Wie jede hard-pruning- Methode hat sie erhebliche unerwünschte Nebenwirkungen und wurde daher im Match gegen Kasparov ausgeschaltet. Aber auch ohne diese Methode widerlegen moderne Programme denselben Zug nicht millionenfach. Wie bereits erwähnt, verwendet die Suche dynamische Information. Schwache Züge werden mit hoher Wahrscheinlichkeit als solche erkannt und mittels soft-pruning mit minimalem Aufwand widerlegt.

Now that computers have reached world-champion level, it is time for chess to become a Drosophila again. Champion-level play is possible with enormously less computation than Deep Blue and its recent competitors use. Tournaments should admit programs only with severe limits on computation. This would concentrate attention on scientific advances.

Für diesen Vorschlag benötigt man gar keine gedrosselte Hardware. Es würde genügen, das Programm mit *Lots of Irritating Superfluous Parentheses* zu schreiben. Ich würde allerdings zur marktüblichen Bezahlung zusätzliches Schmerzensgeld verlangen. Tatsächlich erfüllt das kommerziell wichtige Segment der Brettcomputer und der mobilen Geräte dieses Kriterium. Die in einem Brettcomputer verwendeten Mikrocontroller sind um zwei Größenordnungen langsamer als ein Pentium. Man löst das Problem durch eine radikale Abmagerungskur der Bewertungsfunktion. Die Bewertung wird so weit wie möglich auf Terme erster Ordnung reduziert. Die Position einer Figur bestimmt – unabhängig von der übrigen Brettkonfiguration – ihren Wert.

Terme erster Ordnung können sehr effizient inkrementell berechnet werden; so haben derartige Programme keine Freibauern-Bewertung. Es werden in einer Preprocessing Phase an der Wurzel des Suchbaumes alle potenziellen Freibauern-Felder markiert. Zieht im Laufe der Suche ein Bauer auf eines dieser Felder, wird er als Freibauer bewertet. Tatsächlich können aber durch das Schlagen oder Vorziehen von gegnerischen Bauern zusätzliche Freibauern-Felder entstehen. Diese Ungenauigkeit wird für den Geschwindigkeitsvorteil in Kauf genommen. Ein Wettkampf auf schmalbrüstiger Hardware ist ein Wettkampf in der Kunst des Bitschnitzens und der trickreichen Formulierung von Schachwissen in Termen erster Ordnung. Die ersten beiden Fritz-Versionen bestimmten auf diese Art und Weise Freibauern. Nach einer blamablen Niederlage bei den holländischen Meisterschaften hat Frans Morsch in Fritz3 eine

korrekte Bewertung eingebaut. Laut Epikur ist Glück die Abwesenheit von Schmerz. Nichts ist für einen Schachprogrammierer schmerzhafter als ein abscheulicher Verlustzug bei einem Turnier. Diesen Schmerz zu vermeiden und somit glücklich zu sein ist die Triebkraft seiner Tätigkeit.

Ist Hydra intelligent?

Alle heutigen Top-Programme haben eine komplexe Freibauern-Bewertung, die auch zwischen verschiedenen Gefährlichkeitsstufen von Freibauern unterscheiden kann. Auf Grund der großen Suchtiefen funktioniert die Preprocessing-Methode nicht mehr. Außerdem verschiebt sich mit steigender Rechenkraft die Relation zwischen Wissen und Geschwindigkeit. Je mehr Rechenkraft man zur Verfügung hat, desto mehr Wissen kann man profitabel einbauen.

Der bisherige Endpunkt in dieser Entwicklung ist Hydra. Hydra ist technisch gesehen eine Weiterentwicklung von Deep Blue. Sie verwendet wie Deep Blue spezielle Schach-Hardware. In spezieller Hardware kann die Bewertung weitgehend parallel abgearbeitet werden. Zusätzliches Wissen verlangsamt das Programm nur logarithmisch. Zusätzliches Wissen kostet in Form eines größeren Chips Geld und erhöht exponentiell den Aufwand für die Abstimmung der einzelnen Terme. Ab einer gewissen Komplexität der Bewertung wird das weiße Rauschen nicht mehr beherrschbar. Ein Beispiel für die Komplexität heutiger Programme ist der Beginn der folgenden Partie

Hydra – GM Michael Adams, London 2005.

1. e4 e5 2. Sf3 Sf6 3. Sxe5 d6 4. Sf3 Sxe4 5. d4 d5 6. Ld3 Sc6 7. O-O Le7 8.Te1 Lg4 9. c3 f5 10. Db3 O-O 11. Sbd2 Sa5 Adams Leib- und Magenvariante in der Russischen Verteidigung. Zuletzt hatte er diese Variante gegen Leko in Lineares 2005 auf dem Brett.

12. Dc2 Sc6 13. b4 a6 14. Tb1 eine interessante Neuerung von Hydra in dieser häufig gespielten Variante.

Man dachte bisher, dass 14. a4 verpflichtend ist. Leko spielte daher 14. a4 und 15. Tb1.

14… Ld6 15. h3 Lh5 16. b5 die eigentliche Pointe. An dieser Stelle Begriff auch Adams den Sinn von 14. Tb1. Er versank in langes Grübeln. Nach der scheinbar logischen Fortsetzung 16...ab 17. Txb5 Sa5 18. Txd5 Lxf3 19. Sxf3 Lh2+ 20. Kh2 Dxd5 gewinnt Schwarz die Qualität und Weiß das Spiel. Der weiße Druck ist zu stark. Adams wählte daher das kleinere Übel

16…Sa5 Nach diesem Zug stand er bereits mit dem Rücken zur Wand, im 23ten Zug war die Partie faktisch gelaufen. Von diesem schweren K.O. in der Auftaktpartie hat er sich im Laufe des Matches nicht mehr erholt.

Hydra konnte den schönen Zug 14. Tb1 nur spielen, weil die Bewertungsfunktion erkannte, dass Schwarz das nach 16...ab folgende Qualitätsopfer nicht annehmen kann. Zugleich musste das Programm tief genug rechnen um sowohl dieses Opfer als auch den nach 16. ... Sa5 erzielbaren positionellen Vorteil zu „sehen“.

Die Hydra-Bewertungsfunktion hat aber nicht das Geringste mit menschlichem Schachwissen zu tun. Laut den psychologischen Untersuchungen von de Groot und anderen denken starke Schachspieler in Chunks. Chunks sind bekannte Figurenkonstellationen, quasi Schachwörter. Die Menge der Chunks und ihre Relationen untereinander bilden die Schachsprache. Schachspielen lässt sich als ein Diskurs in dieser Sprache interpretieren. Hydra erkennt keinen einzigen Chunk, die Bewertung basiert hauptsächlich auf der dynamischen Relation zwischen den Figuren. Sie ähnelt der

von H.Opfermann in seinem Buch „Die neue Schachschule“  vorgeschlagenen Methode des Plättchenzählens. Jede Seite legt auf jedes erreichbare Feld ein Plättchen. 10 Plättchen entsprechen dem Materialwert eines Bauern. Opfermann gilt unter Schachspielern als grenzdebiler Spinner. Natürlich spielt kein Mensch Schach, indem er mental Plättchen legt und diese abzählt. Plättchenzählen ist unter dem Namen „Mobilität“ ein bewährtes Computerschach-Bewertungskriterium. J. Schaeffer hat vor seinen Confessiones die Virtutitas der Mobilität gepriesen.

Die Plättchenmethode ist in spezieller Hardware sehr effektiv und relativ einfach zu implementieren. Hydra hat nach 2 Maschinentakten alle Plättchen (inklusive Röntgenstrahl-Plättchen) gelegt und abgezählt. Die Neuerung der Hydra-Bewertung besteht wahrscheinlich darin, dass fast alle schachlichen Konzepte als Funktion über die Mobilität formuliert sind.

Härtere Drogen

Bei einer Mensch-Maschine-Partie werden zwei unterschiedliche Spiele gespielt. Die Menge aller Schachsätze ist viel kleiner als die Menge aller möglichen Schachstellungen. Schachprogramme halten sich nicht an die Sprachgrenzen, sie reden für Menschen unverständliches Zeug. Dies war bei der Konzeption von Hydra Absicht. Eine typische Situation während des Adams-Hydra Matches war: Hydra spielt einen Zug. Der Hydra-Schachexperte GM Lutz rollt die Lippe nach unten und sagt „Hmmmm“. C.D. „Is er schlecht“. GM Lutz: „Hmmm, nicht wirklich“. C.D. „Dann is er super, des bringt den Mickey draus“.

Wie bei jeder anderen Drogensucht lässt die Wirkung von Computerschach über die Jahre nach. Man steht vor der Wahl: Trocken werden oder auf eine härtere Droge umzusteigen. Zur Zeit probiere ich beide Varianten aus. Zum einen wende ich die im Hydra-Projekt erworbenen Fertigkeiten in industriellen Projekten an und gehe – neben meiner Arbeit an Hydra – ab und zu auf einen Computer-Go-Trip.

It wasn't a question of skills, it was a question of basic ideas, and it still is. One of them that comes up very clearly is when you compare how well computers play chess with how badly they play go, in spite of comparable effort having been put in. The reason is that in go, you have to consider the situation, the position...and furthermore, you have to identify the parts--and thats something that isn't really well understood how to do even yet.

Nachdem ich weiß, welcher Aufwand in Computerschach notwendig ist, klingt “comparable effort” sehr abschreckend. Es stimmt nur nicht. Schach hat einige Jahre mehr am Buckel. Es gibt in Go keinen Frans Morsch, der seit 25 Jahren nichts anderes macht, als Schach zu programmieren. Das Go-Pendant zu Fritz ist Many Faces. David Fotland hat nach eigenen Aussagen in den letzten 15 Jahren fünf Mannjahre für Many Faces aufgewandt. Man braucht nur die Benutzeroberfläche von Fritz und Many Faces starten, um den Unterschied an Professionalität feststellen zu können.

Es gibt nichts, was auch nur in Ansätzen mit dem Deep Blue oder selbst dem wesentlichen bescheidenerem Hydra-Projekt vergleichbar wäre. Das einzige professionelle Projekt ist das Reverse-Engineering von Handtalk durch ein nordkoreanisches Team. Nicht ganz die feine Art, aber eine bemerkenswerte softwaretechnische Leistung. Es gibt nur vergleichbar viel bedrucktes Papier. Das kann aber nicht Go spielen. Die Vorraussetzungen, in diesem Konzert der Freunde des mechanischen Steinesetzens als zweite Blockflöte mitspielen zu können, sind also gar nicht so schlecht.

The research of Go programs is still in its infancy, but we shall see that to bring Go programs to a level comparable with current Chess programs, investigations of a totally different kind than used in computer chess are needed.

Auf Grund dieses Einleitungszitats in Martin Müllers Dissertation war es nahe liegend, Suzie zu schreiben. Suzie ist ein Schachprogramm, das Go spielt. Um Erfahrung zu sammeln, wird Suzie zuerst als C-Programm entwickelt. Vorausgesetzt, es funktioniert, und es findet sich wie bei Hydra ein potenter Sponsor, soll Suzie anschließend in Go-Hardware (FPGAs) implementiert werden. Die erste, etwas naive Idee war: Statt Plättchenlegen eine gute Influence-Funktion, ein bisserl hineinsuchen und die Sache ist geritzt. Dieser Ansatz hätte den Vorteil, dass er wie die Mobilität in spezieller Go-Hardware sehr effizient zu implementieren wäre. Wie sich aber bald herausstellte, hat die Sache einen kleinen Haken: Sie funktioniert nicht.

Go ist – für Computer – tatsächlich ein bisserl schwieriger als Schach. Wobei gar nicht so einfach zu sagen ist, was die Schwierigkeit ausmacht. Dass man die Situation, die Position und die einzelnen Teile betrachten muss ist eine "No-Na"-Aussage, die einem nicht besonders weiter bringt.

Einfacher zu beantworten ist die Frage, was kein Problem ist: Der Branchingfaktor. Die Zahl von 361 Zügen klingt furchterregend. Aber bereits simples Alpha-Beta reduziert den Faktor auf 19. Mit den in Schach üblichen Pruning-Methoden ist – ohne die Suchqualität wesentlich zu verschlechtern – ein Faktor 10 möglich. Faktor 10 ist noch immer beträchtlich. Aber wenn man von der Rechenkraft des Hydra-Clusters ausgeht, entspricht dies einer Suchtiefe von 9-10 Plies. Allerdings ist – wie wir noch sehen werden – eine mit Schach in der Effizienz vergleichbare Suche sehr schwer zu realisieren. Aber das ist eine andere Frage, die mit dem Branchingfaktor unmittelbar nichts zu tun hat. Es besteht für mich kein Zweifel, dass auf einem größeren Schachbrett mit 360 Zugmöglichkeiten der Wettlauf Mensch-Maschine schon längst wegen sportlicher Wertlosigkeit abgebrochen worden wäre. Der Mensch würde ständig ein taktisches Detail übersehen.

Ein lösbares, aber ernstzunehmendes Problem ist der Entwurf von effizienten Datenstrukturen und Algorithmen. Das inkrementelle Berechnen von Brett, Ketten und Gruppeninformation ist alles andere als trivial. Die Datenstrukturen müssen – für einen Alpha-Beta–Sucher – reversibel sein. Selbst für einen Menschen einfache Fragestellungen wie „Eine Seite hat ein Gebiet vollständig umschlossen. Liegt Schnittpunkt x innerhalb oder außerhalb dieses Gebietes?“ ist – wenn man auf Effizienz Wert legt – eine harte Nuss. Hinzu kommen noch sehr lästige Sonderfälle am Brettrand und in der Ecke.

Schachprogrammierer haben sich über diese Fragen sehr viele Gedanken gemacht. Es gibt sehr trickreiche

Brettrepräsentation denen eines gemeinsam ist: Die interne Repräsentation ist kein 8x8 Quadrat. Die Go-Literatur behandelt dieses Thema bestenfalls en passant. Die Effizienz von Datenstrukturen und Algorithmen ist im bisherigen Ansatz ein untergeordnetes Problem. Die Programme können mit der Zeit sowieso nichts anfangen. Auf einer unendlich schnellen Hardware würden sie schneller als Lucky Luke ziehen, aber sich genauso wie heute ins Knie schießen.

Bereits ans Eingemachte geht die Frage nach dem inneren Wert der Steine: Ein Stein kann, wenn er tot ist, ein Vorteil für den Gegner sein, aber auch ein toter Stein kann Aij haben, er kann eigenes Gebiet verstellen, er kann aber auch der König in der Stellung sein. Wenn der fällt, bricht die Stellung zusammen. Atari wird manchmal als Schach übersetzt. Es bedeutet aber nur, dass man im nächsten Zug schlagen kann. Die bedrohte Gruppe kann der König oder vollkommen wertlos sein.

Gruppen sind die Figuren des Go-Spiels, wobei in der Regel sehr schwer festzustellen ist, welche Figur überhaupt am Brett steht. Im Schach ist hingegen ein Springer ein Springer. Der kann gut oder schlecht stehen. Wenn er gut steht ist er 3.5, wenn er schlecht steht 3.0 Bauerneinheiten wert. Der schwer bestimmbare Wert von Steinen und Gruppen hat gravierende Auswirkungen auf den Suchprozess. Im Schach ist die einfache Regel „Schlage die größte gegnerische mit der kleinsten eigenen Figur“ ein sehr gutes Kriterium zur Zugsortierung. Wenn die Bewertung für die Seite am Zug 6 Bauerneinheiten schlechter als Alpha ist, kann man am Horizont das Schlagen von Bauern und Leichtfiguren ignorieren. Der Materialwert ist zu gering um in das Alpha-Beta Fenster zu kommen. Natürlich gibt es dazu Ausnahmen, aber die Beschleunigung der Suche ist wesentlich wichtiger als das Restrisiko. Es gibt in Go keine vergleichbar effektive Sortierregel, und auf Grund des unscharfen Materialwertes muss man in der Ruhesuche den Sicherheitsabstand wesentlich größer ansetzen.

Horizonte

Im Lochkarten-Zeitalter, als die Programme typischer weise 5-6 Plies tief suchten, war der Horizonteffekt DAS Problem in Computer Schach. Der Horizont-Effekt existiert auch bei einer 17-Ply Suche, er hat aber seinen ursprünglichen Schrecken verloren. Abgesehen von der tieferen Grundsuche sind Heuristiken wie z.B. die Single-Extensions entwickelt worden, die die wichtigsten Fälle abfangen.

Angenommen, der schwarze Läufer kann entlang der langen Diagonalen den Turm schlagen. Das Programm kann aber durch das Zwischenstellen von Bauern oder Leichtfiguren das Problem über den Suchhorizont schieben. Solange ihm dies gelingt, schätzt es die Stellung besser ein als sie ist. Am Ende gehen nicht nur der Turm, sondern auch Bauern und Leichtfiguren verloren.

Allerdings kosten diese Verzögerungszüge Material. Das Programm wird in eine derartige Stellung nur hineinspielen, wenn es z.B. vorher eine Leichtfigur gewinnen kann. Wenn der Turm ohne vorhergehende Kompensation verloren geht, ist das zusätzliche Opfern von Figuren ohne Auswirkung. Die Stellung ist so oder so verloren.

In Go gibt es hingegen Horizont-Züge die fast nichts kosten. Z.B. das Programm spielt einen Auto-Atari-Zug in das Gebiet des Gegners. Der Gegner muss reagieren indem er den Stein schlägt. Das kostet wenig, weil der Gegner zwar einen Stein bekommt, aber gleichzeitig mit dem Schlagzug einen Gebietspunkt verliert. Meistens kostet es einen kleinen positionellen Faktor, weil der Gegner seine Stellung mit dem Schlagzug stärkt. Auf Grund der geringen Kosten werden nicht nur massive taktische Probleme, sondern auch kleinere positionelle Nachteile über den Horizont geschoben. Wenn das Programm in einer Partie 20 Auto-Ataris ausführt, summieren sich die positionellen Nachteile ordentlich auf. Das Programm jagt dem kurzfristigen Vorteil nach, weil es die langfristigen Konsequenzen leicht verdrängen kann. Diese Auto-Atari-Züge schauen vor allem schwachsinnig aus, und der Programmierer erleidet beim Zusehen Höllenqualen. Klassische Programme kennen kein Horizontproblem. Wenn man keinen Horizont hat, kann man auch nichts über den Horizont schieben.

Suzie hat – wie die klassischen Programme – innerhalb der Bewertung einen lokalen Taktiksolver. Meist gibt es in jeder Stellung mehrere kleinräumige taktische Gemetzel. Die Frage ist nun: Wer ist wo am Zug. Wenn man annimmt, die Seite am Zug ist auch in jedem Subspiel am Zug, dann ist das Zugrecht sehr viel wert. Das Programm macht alles, um am Horizont am Zug zu sein. Wie bereits oben beschrieben hat es auch genügend Möglichkeiten dazu. Man kann bekanntlich aber nur auf einem Kirtag tanzen. Es gibt zwar die Theorie der lokalen Spiele, bis auf Spezialfälle im Endspiel ist diese Theorie aber nur zum Krenreiben gut.

Eine weitere giftige Frage ist das Haifisch-Problem. Eine schwache weiße Gruppe A wird von einer schwarzen Gruppe B angegriffen die ihrerseits von einer noch stärkeren weißen Gruppe C bedroht ist. A lebt wenn C B fressen kann, B lebt, wenn es vorher A frisst. Das kann rekursiv so weitergehen. Es gibt auch kaum Fixpunkte. Selbst eine taktisch tote Gruppe kann noch Einfluss auf das Geschehen haben. Auch wenn die Sache lokal klar ist, kann das globale Ergebnis wegen des Kirtag-Problems davon abweichen. Man muss woanders zuschnappen, um dort noch größeres Unheil zu verhindern.

Eine Alpha-Beta-Suche erzeugt – das ist bis zu einem gewissen Grad auch ihr Sinn – bizarre Stellungen. Die Bewertungsfunktion muss auch diese bizarren Stellungen sinnvoll bewerten. Z.B. kenne ich keine Schachpartie mit einem Quattro-Bauern. In einer tiefen Alpha-Beta Suche muss man aber davon ausgehen, dass so ein Monstrum regelmäßig auftaucht. Wenn man den Quattro-Bauern falsch bewertet, dann findet ihn die Suche und spielt in diese Variante hinein.

In einem konventionellen Go-Programm ist die Bewertung von unterschiedlich langen Leitern kein Problem. Die selektiven Zuggeneratoren schlagen einen Fluchtzug nur vor, wenn es einen Leiterbrecher gibt. Ein Alpha-Beta-Sucher probiert auch einen hoffnungslosen Fluchtzug aus. Wahrscheinlich erkennt die Suche bald die Sinnlosigkeit und verkürzt die Variante. Die Bewertung muss daher die längere Leiter schlechter bewerten. Für diesen Spezialfall ein lösbares Problem, es ist aber praktisch unmöglich, eine perfekte Konsistenz der Bewertung zu erreichen. Schon allein wegen des Konsistenzproblems muss man die Anzahl der Bewertungsterme beschränken.

Diese unvollständige Liste der Probleme klingt abschreckend. Dem steht aber eine einzige sehr einfache Tatsache gegenüber. Die Suche ist in Go mindestens genauso wirksam wie in Schach. Jeder zusätzliche Ply bringt mindestens genauso viele Elopunkte wie im Schach. Die Suzie-Bewertungsfunktion wurde in letzter Zeit von Peter Woitke, dem Autor des klassischen Go-Programms GoAhead, gründlich überarbeitet und verbessert. Sie ist trotzdem wesentlich einfacher – und wesentlich schneller – als die von Gnu-Go. Die Suche ist eine vereinfachte Version des Hydra-Codes. Sie enthält keinerlei Go-spezifische Heuristiken. Go-Wissen geht nur über die Bewertung in die Suche ein. In der aktuellen Version liegt der Break-Even-Point zwischen 5 und 6 Plies. Auf einem 9x9 Brett erreicht Suzie bei einer Zeitkontrolle von 5 Minuten/Partie typischer Weise Tiefe 7. Gnu-Go wird dementsprechend auch klar geschlagen. Bei längerer Zeitkontrolle sind die Partien bereits sportlich wertlos. Auf 13x13 halten sich die Programme die Waage, auf 19x19 ist Gnu-Go noch Herr der Lage. Auf 19x19 ist die PC-Hardware für einen Alpha-Beta Sucher noch zu schwach (und Suzie noch zu wenig ausgereift). Das Problem ließe sich mit Hilfe eines Suzie-Scheiches auch mit heutiger Technologie lösen. Auf dem Hydra-Cluster würde Suzie auch auf 19x19 tief genug rechnen. Es gibt keinerlei prinzipiellen Grund, warum ein Schachprogramm, das Go spielt, nicht funktionieren sollte. Wir sind nur nicht im Computer-Schachjahr 2007 oder 1989, sondern im Jahr 1970.

Das Gleichgewicht der Dummheit

GM Lutz hat für Endspielstellungen untersucht welchen Einfluss ein Mehrbauer auf die Gewinnverteilung hat. Der Wert des Bauern hängt von den übrigen Figuren, der Anzahl der Bauern und ihrer Verteilung auf den Flügeln ab, sie ist jedoch weitgehend unabhängig von der Elozahl der Spieler. Für ein bestimmtes Endspiel ist es egal, ob es zwei GMs oder zwei Klubspieler gegeneinander spielen. Die Klubspieler machen mehr Fehler, aber diese Fehler gleichen sich aus, und am Ende ist der Mehrbauer genauso viel wert wie zwischen zwei Könnern. Möglicherweise würde auch eine Partie zwischen zwei Schimpansen eine ähnliche Gewinnverteilung aufweisen. Das ist die Grundidee von Monte-Carlo-Go. Eine Stellung wird bewertet, indem man Zufallspartien bis zum Ende spielt. Die Bewertung der Stellung ist der Prozentsatz der gewonnen Partien. Im einfachsten Fall kennt ein Monte-Carlo Programm nur die Go-Regeln. Zusätzlich spielt es nicht in ein eigenes Ein-Punkt Auge. Es wird nach den chinesischen Regeln gnadenlos solange gespielt, bis nur mehr eigene Einpunkt-Augen übrig bleiben. Die Partien sind aus menschlicher Sicht bizarr.

Man kann nun in der Ausgangsposition alle Züge ausführen und pro Zug n Monte-Carlo Partien durchführen. Der Zug mit den meisten Gewinn-Partien wird gespielt. Aus der Theorie der Multi-Armed-Bandits sind bessere Methoden bekannt. Es ist effektiver den Aufwand auf die aussichtsreichsten Züge zu konzentrieren. Zu Beginn der Suche haben alle Züge das gleiche Gewicht und werden der Reihe nach durchprobiert. Erfolgreiche Züge erhalten ein stärkeres Gewicht und werden öfter ausprobiert als schlechte. Wobei auch schlechte Züge nicht gänzlich ausgeschieden, sondern mit geringer Wahrscheinlichkeit weiter durchgerechnet werden. Der Zug könnte bisher nur Pech gehabt haben. Die Grundidee ist analog zum Soft-Pruning in der modernen Alpha-Beta Suche.

Die Methode lässt sich rekursiv verallgemeinern. Man kann auch den besten Gegenzug auf dieselbe Weise bestimmen. Und den Gegenzug auf den Gegenzug…

Diese Methode wurde UCT (*Upper Confidence bounds applied to Trees*) getauft. Der Pionier auf diesem Gebiet ist Remi Coulom mit seinem Programm Crazy Stone. Remi hatte zuvor das gute Amateur-Schachprogramm Crazy Bishop entwickelt. Anstatt vor den Mühen einer brauchbaren Go-Bewertung zu resignieren, hat er mit diesem wissensfreien Ansatz einen großen Schritt nach vorne getan. Noch erfolgreicher als Crazy-Stone ist MoGo von Sylvain Gelly. Beide Programme spielen nicht gänzlich zufällig. Die Züge werden auch in der Monte-Carlo Phase gewichtet. Die Regeln dafür sind aber immer noch auf Schimpansen Niveau. Z.B. wenn man eine Gruppe schlagen kann, dann soll man sie schlagen. Es ist gut in der Nähe des letzten gegnerischen Zuges zu spielen. Wie bei Alpha-Beta erweist sich auch bei UCT das Feintuning der Suche als äußerst wichtig. Nach einer Untersuchung des ehemaligen professionellen Schachprogrammierers Don Dailey skaliert UCT besser als die Alpha-Beta-Suche in Schach. Eine Verdoppelung der Monte-Carlo Versuche verbessert ein Programm um fast 200 Elo. In Schach muss man dafür einen Ply tiefer rechnen oder mit anderen Worten den Aufwand vervierfachen.

Meiner Meinung ist das Dogma der Go-Programmierung: „eine Suche bringt nichts“, längst widerlegt. Die Suche ist in Go genauso erfolgreich wie in Schach. Es ist nur die Frage, welche Suche: Modernes Alpha-Beta oder UCT. Im Moment hat UCT die besseren Karten, aber man sollte Alpha-Beta nicht gänzlich wegprunen. Möglicher Weise ist auch eine Kombination beider Methoden möglich.

John McCarthy hatte Recht. Im Go sind gänzlich andere Methoden notwendig. Nur hat er mit Sicherheit nicht UCT gemeint. UCT ist DIE Antithese zum Drosophila-Ansatz. Kein Mensch spielt auf diese Weise Go. Parallelen gibt es hingegen zum Verhalten eines Ameisenhaufens. Die Entwicklung eines Nano-Chips, um das Verhalten von Ameisen beim Bau von Ameisenstrasse zu verbessern, gehört wahrscheinlich nicht zu den Kernaufgaben der AI. Wenn die AI-Zunft halbwegs konsequent ist, muss sie nun auch Go den Status als Drosophila absprechen. Für Schach war die Ungnade sehr produktiv. Persönlich finde ich die Vertreibung aus dem Paradies betrüblich. Als ich die ersten Überlegungen für Suzie anstellte, bin ich von einer auf niedrigem Niveau erstarrten Disziplin ausgegangen. Nun hechelt das Suzie Team (Peter Woitke, Stefan Mertin und C.D.) hinterher.

Fazit

Über die interessante Frage, warum die Ökonomen und die AI-Jünger systematisch falsch liegen, kann man nur spekulieren. Eine mögliche Erklärung ist: Es handelt sich nicht um Wissenschaft sondern um ein als Wissenschaft verkleidetes Religions-Surrogat. Das Ideal der Ökonomen ist ein von Sitte, Anstand und Moral unbefleckter Nomade, der nur ein Credo kennt: die Maximierung des ökonomischen Erwartungswertes. Fleisch gewordenes Kapital. Die Gleichungen der Ökonomen funktionieren nur unter dieser Voraussetzung. Eine freie Gesellschaft zeichnet sich dadurch aus, dass Arme und Reiche gleichermaßen die Freiheit haben, unter der Brücke zu schlafen. Gott beging einen katastrophalen Fehler, als er die Welt ohne externen Consultant erschuf. Wie konnte er am siebten Tage ruhen? Wegen des schlechten Vorbildes sitzen eine Milliarde Menschen vollkommen unproduktiv in Kirchen, Moscheen und Synagogen herum. Sieht man einmal von der unbedeutenden Branche der Kirchenwirte ab, entsteht dadurch ein gigantischer gesamtwirtschaftlicher Schaden!

John McCarthy sieht sich hingegen als Prophet der schönen neuen Techno-Welt, dessen Aufgabe es ist den siebten Tag der Schöpfung nachzuholen. Man könnte es als harmlose Frohbotschaft betrachten, die nichts bringt, wegen ihrer praktischen Sterilität aber auch keinen Schaden anrichtet. Bei der Vorbereitung dieses Artikels habe ich mich auf McCarthys Homepage umgesehen. Ich war auf der Suche nach Go-Zitaten. Aus Neugier habe ich auch den Verweis "Sustainable" angeklickt. Unter dieser - etwas irreführenden - Überschrift stellt sich McCarthy unter anderem die Frage:

Won't global warming do us in unless we drastically reduce our use of energy?

und gibt die Antwort:

No. Global warming can be avoided or reversed should it turn out to be a serious problem. However, there is a thorough paper Why Global Warming Would be Good for You by Tom Moore of the Hoover Institution. It is still controversial whether global warming from CO2 is occurring or whether recent warm years are a statistical fluctuation or a consequence of changes in the sun.

Here is Health and Amenity Effects of Global Warming, also by Tom Moore. It offers statistical evidence that regions of the U.S. with warmer climates have lower death rates and also are preferred to colder regions.

Die Klimaerwärmung als Mittel zur Kreislaufvorsorge. Als gelernter Statistiker erinnert mich das Herzkasperl-Argument an die statistischen Studien, die einen klaren Zusammenhang zwischen dem Rückgang der Storchenpopulation und der Geburtenrate nachweisen. Die Zahlen sind eindeutig, die Interpretation, dies sei der Beweis, dass die Kinder von den Störchen gebracht werden, aber doch etwas gewagt.

McCarthys Aussage könnte aus einer bitterbösen Satire von Michael Moore über die Bush-Administration stammen. Wie ich von McCarthy nahe stehenden Kreisen unterrichtet wurde, versteht er in diesen Zusammenhang keinen Spaß. Nach dem Entropiesatz folgt daraus: Wir haben ein Problem. Man kann auch ohne das Studium seiner Homepage zu dieser Meinung gelangen. Es ist aber doch erhellend, den UN-Klimabericht kurz und prägnant bestätigt zu bekommen.