Lösung eines alten Problems

Ein neues Verfahren zur Auswertung von Stellungstests

Die Frage nach der korrekten Auswertung von Stellungstests ist fast schon so alt wie das Computerschach selbst. Manchmal genügt ein ungewöhnlicher Blick auf altbekannte Tatsachen, um zu einer überraschend einfachen Antwort zu gelangen. Im vorliegenden Fall führte dies zu einem neuen Ansatz für die Auswertung von Stellungstests, oder sagen wir vielmehr, zu einem fast neuen Ansatz. Aber lesen Sie selbst...

Zur genauen und zuverlässigen Ermittlung der praktischen Spielstärke eines Schachprogramms sind meist sehr viele Partien gegen eine möglichst große Anzahl verschiedener Gegner notwendig. Ein Blick auf die einschlägigen Ranglisten wie die schwedische SSDF- oder die Best-for-Fritz(BfF)-Rangliste zeigt, dass viele hundert Partien mit einem Programm absolviert werden müssen, um seine Spielstärke mit einer Genauigkeit von mindestens 30 Elopunkten angeben zu können.


Dr. Frank Schubert

Die modernen Benutzeroberflächen wie die der Fritz-Familie, aber auch die Chess Partner-, Chess Assistant-, Winboard- oder Arena-GUI erlauben es, diese Sisyphusarbeit nahezu vollautomatisch und unter fest definierten Bedingungen auf ein und demselben Rechner durchzuführen. Dies war jedoch nicht immer der Fall. In den Frühzeiten des Computerschachs, insbesondere in der Blütezeit der Brettcomputer, musste jede Partie noch mühsam per Hand ausgetragen werden. Aus diesem Grunde begann man damals, nach alternativen und weniger zeitintensiven Methoden zur Spielstärkeermittlung Ausschau zu halten. Dies war die Geburtsstunde der Stellungstests, bei denen jedes Programm über einen vorher festgelegten Parcours von taktischen oder positionellen Stellungsaufgaben geschickt wird, um nachher aus dem Lösezeitverhalten des Programms auf seine praktische Spielstärke schließen zu können.

Wohl kein anderer Stellungstest hat die Gemüter der Computerschach-Freunde mehr erregt, als der im Jahre 2001 in der CSS von den Autoren Gurevich und Schumacher vorgestellte WM-Test, der in seiner ersten Fassung 40 und nach verschiedenen Ausbaustufen nunmehr insgesamt 100 Stellungen zu den drei Themengebieten Königsangriff, Positionsspiel und Endspiel enthält (siehe CSS 5+6/2001 sowie 1, 2+4/2002). Im Internet-Forum der CSS gab es in den vergangenen Monaten zahlreiche Diskussionen, die sich meist sachlich, teilweise aber auch sehr emotional mit Sinn und Unsinn dieses Tests und von Stellungstests im Allgemeinen auseinander gesetzt haben.

Die Kritik orientierte sich dabei vor allem an der Beobachtung, dass die WM-Test-Bewertung einzelner Programme teilweise eklatant von ihrer im praktischen Spiel erworbenen Elozahl abweicht. Auch wurde darauf hingewiesen, dass die Elodifferenz zwischen den besten und schlechtesten Programmen im praktischen Spiel um ein Vielfaches größer ist als beim WM-Test. Diese Tatsachen verleitete die Autoren dann schließlich zur Einführung des Begriffs der so genannten "Analysespielstärke", die nicht notwendigerweise mit der tatsächlichen Spielstärke im Partieschach übereinstimmen, aber dennoch eine gewisse Korrelation zu dieser aufweisen sollte. Die im WM-Test ermittelte Wertungszahl wurde somit zwar gewissermaßen unangreifbar gegen alle Kritik, sie machte aber gleichzeitig auch den Stellungstest im Grunde genommen überflüssig, da sie dessen ursprüngliches Ziel, nämlich die Vorhersage der praktischen Spielstärke aus den Augen verlor.

Um letzteres Ziel zu erreichen, muss ein Stellungstest drei wesentliche Bedingungen erfüllen:

- Er muss über ein fundiertes, zuverlässiges und allgemein anwendbares Auswerteverfahren verfügen.
- Er muss eine ausgewogene und für das praktische Spiel repräsentative Auswahl von Teststellungen bereitstellen.
- Er muss eine ständige Anpassung an die immer weiter voranschreitende Hardware- und Softwareentwicklung erlauben, ohne die Notwendigkeit einer völligen Neukonzeption ("dynamischer Test").

Wir wollen uns in dem vorliegenden Beitrag zunächst auf den ersten Punkt, nämlich das Auswerteverfahren konzentrieren, da hier nachweislich die größten Defizite liegen.

Lösung eines alten Problems

Existierende Stellungstests

Tabelle 1 gibt eine Übersicht der wichtigsten Stellungstests der vergangenen Jahre, ihrer Autoren sowie der zugrunde liegenden Auswerteformel. Das größte Manko dieser Formeln besteht insbesondere darin, dass sie keinerlei fundierte mathematische oder schachliche Basis haben und lediglich durch "Ausprobieren" gefunden wurden. Sie weisen schwere systematische Mängel auf und sind alles in allem schlichtweg falsch, was sich recht einfach nachweisen lässt.

Name, Jahr	Autoren	Auswerteformel	Bemerkungen
BS-2830, 1997	Bednorz, Schumacher	$BS = 2830 - (GZ / 17)^2$	27 Stellungen a 15 min
BS-2830, 1998	Bednorz, Schumacher	$BS = 2830 - GZ / 1{,}5 - (GZ/22)^2$	"verbesserte" Formel
GS-2930, 2000	Gurevich, Schumacher	$GS = 2930 - 1{,}5 \times GZ$	14 Stellungen a 20 min
HB-2950, 2001	Hörnig, Bednorz	$HB = 2950 - 2 \times GZ / 5 - (GZ / 50)^2$	Studientest, 30 Stellungen a 30 min
BS-2001, 2001	Bednorz, Schumacher	$BS = 2700 + 2 \times LQ - GZ / 2$	30 Stellungen a 15 min
WM-Test, 2001	Gurevich, Schumacher	$GS = 2600 + 2 \times LQ - 5 \times GZ / N_0$	$N_0 = 40$ bzw. 100 Stellungen a 20 min
Scheidl-Test, 2002	Scheidl	$Elo = BW + 240 \times LQ + 60 \times MW[\log_2(MZ / LZ)]$	BW: Elo-Basiswert MW[...]: Mittelwert [...] MZ: Maximalzeit LZ: Lösezeit

Tabelle 1: verschiedene Stellungstests mit zugrunde liegender Auswerteformel. Dabei ist jeweils GZ die Gesamtlösezeit (in sec) und LQ die Lösequote, also die Zahl der gelösten Stellungen divitiert durch die Gesamtzahl aller Stellungen (N_0). In der Scheidl-Formel ist der Basiswert BW beliebig. Die Maximalzeit MZ ist die vorher festgelegte Wartezeit je Stellung (also z.B. 20 min). LZ ist die Lösezeit der *einzelnen* Stellung. Der Mittelwert erstreckt sich bie Scheidl über alle untersuchten Stellungen.

So beinhalten die Auswerteformeln von BS-2830, GS-2930, HB-2950 und BS-2001 als Variable die absolute Gesamtlösezeit des Tests, nämlich GZ. Sie sind daher ausschließlich für diesen einen Test konzipiert und nutzbar. Sobald andere Stellungen verwendet werden oder die Anzahl der Stellungen verändert wird, werden diese Formeln unbrauchbar. Erst der WM-Test führte erstmalig eine von den Teststellungen unabhängige Formel ein.

Die ermittelte Wertungszahl hängt je nach Test und Formel quadratisch oder linear von der Gesamtlösezeit ab, was zu unsinnigen Ergebnissen führt. Nehmen wir als Beispiel die Formel des GS-2930-Tests aus dem Jahre 2000 und betrachten zwei Programme, für die eine Gesamtlösezeit von 240 bzw. 120 Sekunden ermittelt wurde. Nach der zugrunde gelegten Auswerteformel ergeben sich die Wertungszahlen GS = 2570 und 2750, was einer Wertungszahldifferenz von 180 Punkten entspricht. Dies steht im Widerspruch zu der Erfahrung aus den Eloranglisten, bei denen eine Verdopplung der Rechengeschwindigkeit des verwendeten PCs (und somit eine Halbierung der Lösezeit im Stellungstest) mit einem Elozugewinn von lediglich 70 Punkten einhergeht.

Stellen wir uns weiter vor, dass beide Programme auf einem Rechner getestet werden, der nur noch halb so schnell ist wie der erste. In diesem Fall würde man erwarten, dass sich zwar die absoluten Elozahlen verändern, die Elodifferenz zwischen beiden Programmen sollte aber konstant bleiben. Stattdessen erhält man mit den verdoppelten Lösezeiten von 480 bzw. 240 Sekunden die neuen Wertungszahlen GS = 2210 und 2570, also eine Differenz von 360 Punkten. Dies bedeutet, dass die Spielstärkeunterschiede von der Geschwindigkeit des Rechners abhängen, auf dem der Test durchgeführt wurde. Dies ist natürlich mehr als unbefriedigend.

Als erstes Fazit obiger Überlegungen lässt sich daher festhalten, dass jede der alten Auswerteformeln, in die statt einem Verhältnis von Lösezeiten die absoluten Lösezeiten der Programme eingehen, zwangsläufig, wie oben beschrieben, zu falschen Ergebnissen führen muss. Darüber hinaus sollte eine korrekte Formel der Tatsache Rechnung tragen, dass eine Halbierung der Lösezeit eines Programms einen Elozugewinn von ca. 70 Punkten bewirkt. Ein solcher Zusammenhang kann mathematisch mithilfe einer logarithmischen Formel ausgedrückt werden, die Scheidl erstmalig in seinem Test von 2002 einführt (mit dem Unterschied, dass er mit einem Elozugewinn von 60 statt 70 Punkten arbeitet, siehe CSS 2/2002).

Auch auf einen weiteren wichtigen Punkt macht Scheidl aufmerksam. So weist er nämlich daraufhin, dass die Auswertung einer Gesamtlösezeit zu unlogischen Ergebnissen führen kann, da ein und dieselbe Lösezeit auf verschiedene Art und Weise zustande kommen kann. Werden etwa von einem Programm A neun Stellungen jeweils in 5 sec und eine zehnte Stellung in 155 sec gelöst, so erhält man eine Gesamtlösezeit von 200 sec. Dieselbe Summe erhält man aber auch, wenn von einem Programm B alle zehn Stellungen in exakt 20 sec gelöst werden. Sind die Programme A und B daher gleich stark? Offensichtlich nicht, da Programm A im direkten Stellungsvergleich neunmal die Nase vorn hat und nur ein einziges Mal deutlich schlechter abschneidet. Eine Auswerteformel, die diese Tatsache berücksichtigt, sollte also jede Stellung einzeln betrachten, anstatt von einer Gesamtlösezeit über alle Stellungen auszugehen.

Die beiden oben aufgeführten Punkte sind zwar in der Scheidlschen Formel weitestgehend realisiert, letztere weist aber dennoch zwei bedeutsame Schwächen auf. Ein erster Fehler liegt darin, dass im Argument der Logarithmusfunktion die maximale Wartezeit je Stellung, MZ, eingeht. Dies führt zu dem kuriosen Ergebnis, dass die Wertungszahl eines Programms, das alle Stellungen des Tests innerhalb der Wartezeit löst (also eine Lösequote von 100 % aufweist) direkt

von eben dieser Wartezeit abhängt, was natürlich nicht korrekt ist. Stattdessen sollte die Elozahl in diesem Fall konstant bleiben, wenn die Wartezeit immer weiter vergrößert wird.

Eine zweite Schwäche ist die Einbeziehung der Lösequote LQ. Sie wurde erstmalig im BS-2001-Test eingeführt, mit dem durchaus begrüßenswerten Ziel, einen weiteren Indikator zur leistungsmäßigen Unterscheidung der Programme im Stellungstest zu erhalten. Das große Problem bei diesem Ansatz besteht jedoch darin, dass die Lösequote keine von den Lösezeiten unabhängige Größe darstellt. Ungelöste Stellungen könnten ebenso gut über die Maximalzeit Eingang in das Auswerteverfahren finden. Überdies ist der Zusammenhang zwischen der Lösequote und der Eloleistung eines Programms völlig unklar. Aus diesem Grunde stellt die Lösequote keinen geeigneten Gradmesser zur exakten Erfassung der Spielstärke eines Programms dar.

Aus der obigen Erörterung geht eindeutig hervor, dass bislang kein theoretisch fundiertes, zuverlässiges und universell einsetzbares Auswerteverfahren für Stellungstests existiert. Im nächsten Kapitel wird daher eine Methode vorgestellt, welche die Schwächen der bisherigen Formeln beseitigt und erstmalig auf einer soliden schachlichen Theorie basiert.

Lösung eines alten Problems

Ein neuer Ansatz

Die Hauptidee besteht darin, den Vergleich der Lösezeiten zweier Programme A und B für eine *einzelne* individuelle Teststellung als eine Art schachlichen Wettkampf zwischen diesen beiden Programmen aufzufassen. Das Programm mit der besseren Lösezeit (LZ) gewinnt diesen Miniwettkampf. Ähnlich wie beim normalen Partieschach könnte man im einfachsten Fall eine 1-0-, 0-1- oder Remis-Wertung einführen, je nachdem, ob der Stellungsvergleich gewonnen oder verloren wurde (bessere bzw. schlechtere LZ), oder unentschieden endete (gleiche LZ). Allerdings würde man damit eine Menge Zusatzinformationen leichtfertig aus der Hand geben, denn die beiden Lösezeiten beinhalten nicht nur die Aussage, welches Programm besser abgeschnitten hat, sondern auch, um wie viel besser es war.
Diese wichtigen Informationen nutzen wir nun dadurch aus, dass abhängig vom Verhältnis der beiden Lösezeiten t_1 und t_2 jedem Stellungsvergleich ein eindeutiges Ergebnis zwischen 0 und 1 zugewiesen werden kann, also z.B. 0,2–0,8 oder auch 0,61–0,39 etc. Wie gelangen wir aber nun zu einem Ausdruck, der diese Werte bereitstellt und gleichzeitig dafür sorgt, dass eine Halbierung der Lösezeit mit einem Elozugewinn von 70 Punkten einhergeht? Der gesuchte Zusammenhang kann durch einen logarithmischen Ausdruck realisiert werden. Danach gilt für die Elodifferenz D eines Stellungsvergleichs:

$$D = 70 \log_2\left(\frac{t_2}{t_1}\right)$$

Formel (1)

Wir bilden also zunächst das Verhältnis der beiden Lösezeiten t_2 und t_1, berechnen davon den Logarithmus zur Basis 2 und multiplizieren das Ergebnis anschließend mit dem konstanten Elowert 70. Dass diese Formel die oben formulierten Anforderungen erfüllt, kann man schnell nachprüfen. Der Zweier-Logarithmus liefert für das Lösezeitverhältnis $t_2/t_1 = 2$ den Wert +1 und für $t_2/t_1 = ½$ den Wert –1. Multipliziert mit 70 erhalten wir somit eine Elodifferenz von +70 bzw. –70 Punkten, je nachdem, ob t_2 doppelt so groß oder halb so groß ist wie t_1. Sind beide Lösezeiten gleich groß, so wird der Logarithmus und damit auch die Elodifferenz null.
Damit sind wir aber noch nicht am Ziel. Wir brauchen nun noch einen Ausdruck, der die Elodifferenz in ein prozentuales Ergebnis (Score) zwischen 0 und 1 überführt. Dazu verwenden wir die bekannte Eloformel (siehe CSS 1/2000), nach der für den Score p gilt:

$$p = \frac{1}{1+10^{-D/400}}$$

Formel: (2)

Setzt man Gleichung (1) in den Ausdruck (2) ein, so ergibt sich nach einigen elementaren Umformungen die folgende einfache Formel, die jedem individuellen Stellungsvergleich zwischen zwei Programmen mit den Lösezeiten t_1 und t_2 ein eindeutiges Ergebnis zwischen 0 und 1 zuordnet:

$$p = \frac{1}{1+\left(\frac{t_2}{t_1}\right)^{-0.5813374}}$$

Formel: (3)

In Tabelle 2 sind einige Beispiele aufgeführt, die verdeutlichen, welche Resultate sich in Abhängigkeit von den beiden Lösezeiten ergeben können.

Lösezeit t_1 in sec	Lösezeit t_2 in sec	Score p	1 - p
40	20	0,40	0,60
20	40	0,60	0,40
40	80	0,60	0,40
80	80	0,50	0,50
80	2	0,10	0,90
8000	2	0,01	0,99

Tabelle 2: Ergebnis des Stellungsvergleichs aös Funktion der Lösezeiten t_1 und t_2 gemäß Formel (3).

Im Idealfall sind beide Lösezeiten bekannt. In der Praxis wird aber auch der Fall eintreten, dass ein Programm oder sogar beide Programme innerhalb der festgelegten Wartezeit die Lösung nicht finden. Gilt dies für beide Programme, so haben wir keinerlei Information darüber, welches Programm in dieser konkreten Stellung besser abschneidet. Somit kann solchen Stellungsvergleichen kein Ergebnis zugeordnet werden.
Ist dagegen bei einem Programm A die Lösezeit bekannt und bei dem anderen Programm B nicht (da es die Stellung innerhalb der Wartezeit nicht lösen kann), so wissen wir zumindest, dass A den Stellungsvergleich gewinnt. Allerdings lässt sich kein exakter Wert als Ergebnis angeben. Was wir aber auf jeden Fall berechnen können, sind eine obere und eine untere Schranke für den Stellungsvergleich. Dieses Prinzip wollen wir an einem konkreten Beispiel nachvollziehen. Nehmen wir an, das Programm A löst eine bestimmte Stellung in 34 Sekunden, Programm B dagegen kann innerhalb der Wartezeit von 300 Sekunden keine Lösung finden. Verlängern wir nun die Wartezeit, so wird die Stellung von B in dem für A schlechtesten Fall kurz danach, z.B. in 300,01 Sekunden, gelöst. Setzen wir nun $t_2 = 300{,}01$ und $t_1 = 34$ in die Formel (3) ein, so erhalten wir als untere Schranke aus der Sicht von Programm A ein Ergebnis von 0,78–0,22. In dem für A günstigsten Fall wird die Stellung dagegen von B überhaupt nicht gelöst, d.h. in einer quasi unendlich langen Zeit. Für diesen Grenzfall ergibt sich aus der Sicht von Programm A eine obere Schranke für das Ergebnis des Stellungsvergleichs von 1–0.
Wir haben damit das unbekannte tatsächliche Ergebnis zunächst auf ein vergleichbar kleines Intervall zwischen 0,78 und 1,0 eingegrenzt. Für unser Auswerteverfahren benötigen wir aber einen konkreten Wert. Ohne zusätzliches A-priori-Wissen über die beteiligten Programme oder weitere statistische Annahmen ist die beste Schätzung für den gesuchten Wert einfach der Mittelwert aus oberer und unterer Schranke. Im vorliegenden Fall ergibt sich demnach als Ergebnis des Stellungsvergleichs ein 0,89–0,11. Untersuchungen an realen Stellungstests mit veränderter Wartezeit haben gezeigt, dass die oben beschriebene Vorgehensweise der Wirklichkeit in der Tat sehr viel näher kommt als alle alten Auswerteformeln, bei denen jedem Programm für eine ungelöste Stellung die Wartezeit zur Gesamtlösezeit hinzuaddiert wurde. In gewissem Sinne ging man dabei also von der unteren Schranke aus und nicht vom realistischeren Mittelwert. Tabelle 3 demonstriert beispielhaft einige mit dem oben geschilderten Verfahren berechnete Stellungsergebnisse. Man erkennt, dass dem stärkeren Programm in jedem Fall ein Ergebnis von 0,75–0,25 sicher ist.

Lösezeit t1 in sec	**Lösezeit t2 in sec**	**Untere Schranke für p**	**Obere Schranke für p**	**Mittelwert für p**
34	>300	0,78	1,00	0,89
1,2	>300	0,96	1,00	0,98
150	>300	0,60	1,00	0,80
299	>300	0,50	1,00	0,7

Tabelle 3: Ergebnisse des Stellungsvergleichs für den Fall, dass eine der beiden Lösezeiten nicht bekannt ist.

Wir verfügen nun also über ein Verfahren, das es erlaubt, jedem einzelnen Stellungsvergleich zwischen zwei Programmen ein eindeutiges Ergebnis zuzuordnen. Diese Prozedur kann für alle Programme und sämtliche untersuchten Teststellungen durchgeführt werden. Bis auf die Tatsache, dass das Ergebnis auch Werte zwischen 0, ½ und 1 annehmen kann, unterscheidet sich das Vorgehen in keinster Weise vom normalen Partieschach. Dies bedeutet, dass das im Programm EloStat realisierte und etablierte Auswerteverfahren vollständig übernommen werden kann.
EloStat wurde in den CSS-Ausgaben 1 und 2/2000 ausführlich vorgestellt. Es steht in der Version 1.2 auf der Internetseite der CSS zum Download zur Verfügung. EloStat benutzt das Iterationsverfahren zur Berechnung einer Elo-Rangliste, indem die zugrunde gelegte PGN-Datenbank als ein einziges großes Turnier aufgefasst und ausgewertet wird. Bereits bekannte Elozahlen der beteiligten Programme werden nicht benötigt.
Basierend auf den oben geschilderten Grundlagen wurde das Programm "EloStat-TestSuite" (im folgenden kurz "EloStatTS") zur vollautomatischen Analyse von Stellungstests entwickelt. Es wird im folgenden Abschnitt beschrieben.

Lösung eines alten Problems

EloStatTS

Das derzeit in der Version 1.0 vorliegende Programm wertet Testsuites aus, die im PGN-Format gespeichert sind. Einzelne Teststellungen müssen in der folgenden Form vorliegen:

```
[Event "952 ECM"]
[Site "Dresden"]
[Date "2003.02.01"]
[Round "?"]
[White "Tactics Marathon"]
[Black "?"]
[Result "*"]
[Annotator "Schubert"]
[SetUp "1"]
[FEN "2k4r/1pp2ppp/p1p1bn2/ 4N3/1q1rP3/2N1Q3/PPP2PPP/ R4RK1 w KQkq - 0 1"]
[PlyCount "1"]

{'Fritz 8-128MB/AMD Athlon 700' 1.35s / 7, 'Chess Tiger 14.0-96MB/AMD Athlon 700' 1.25s / 7
BEST, 'Shredder 7-128MB/AMD Athlon 700' 4.01s / 8} 1. Nd5
```

Die einzige für die Auswertung wichtige Zeile im PGN-Header ist der FEN-String (siehe CSS 5/2001), der zur Identifikation der Stellung verwendet wird. Der Lösezug, in diesem Fall Sd5, muss als erster Zug in der Notation erscheinen. Die Lösezeiten der verschiedenen Programme werden zwischen den geschweiften Klammern {...} im Kommentar angegeben. Zunächst kommt der Programmname in Anführungszeichen '...', dann folgt die Lösezeit (ganze Zahl oder Dezimalzahl) gefolgt von der Rechentiefe hinter dem Schrägstrich. Einzelne Programmeinträge sind durch ein Komma voneinander getrennt.
Die obige Form entspricht genau dem Format, welches die aktuelle Fritz-GUI bei der Auswertung von Stellungstests erzeugt. Sie haben also die Möglichkeit, Ihren bevorzugten Stellungstest zunächst von Fritz & Co abarbeiten zu lassen und danach EloStatTS zur vollautomatischen Auswertung heranzuziehen. Dazu gehen Sie folgendermaßen vor:
Starten Sie Fritz und wählen Sie die Funktion *Extras/Analyse/Testsuite* lösen. Wählen Sie dann die Testdatenbank aus, die im CBH-Format vorliegen muss. Anschließend geben sie die maximale Lösezeit ein und wählen unter Extra-Halbzug mindestens den Wert 2. Danach wird der Test mit dem gerade geladenen Programm ausgeführt. Die Ergebnisse werden automatisch in der CBH-Datenbank abgelegt. Nach Beendigung des Tests müssen Sie diese noch in das PGN-Format konvertieren. Dazu wählen Sie im Datenbankfenster *Datei/ Neu/Datenbank* und speichern eine neue Datenbank mit dem Dateityp PGN ab. Danach wechseln Sie in die ursprüngliche CBH-Datenbank, wählen *Bearbeiten/Alle auswählen* und anschließend *Bearbeiten/Kopieren*. Zu guter Letzt wechseln Sie wieder in die noch leere PGN-Datenbank und wählen *Bearbeiten/Einfügen*. Sämtliche Stellungen mitsamt der Lösezeiten werden nun ins PGN-Format umgewandelt.
Nun starten Sie EloStatTS mit einem Doppelklick auf die Exe-Datei, die sich am besten im selben Verzeichnis befindet, wie die Datenbank selbst. Es erscheint folgender Startbildschirm:

```
EloStatTS, Version 1.0
by Dr. Frank Schubert, 01/2003
---------------------------

Name of pgn-file = marathon
Start Elo = 2600
```

Sie werden zuerst nach dem Namen der PGN-Datenbank gefragt, die ausgewertet werden soll. In diesem Fall wurde "marathon" (ohne PGN-Endung!) eingegeben und mit *Enter* bestätigt. Anschließend geben Sie den Elostartwert ein (in diesem Fall 2600) und bestätigen erneut mit *Enter*. Der Rechenvorgang wird jetzt gestartet und kann am Bildschirm verfolgt werden:

```
Reading marathon.pgn...
(1) Fritz 8-128MB/AMDAthlon700
(2) Chess Tiger 14.0-96MB/AMDAthlon700
(3) Shredder 7-128MB/AMDAthlon700
(4) Hiarcs 8-128MB/AMDAthlon700
(5) Junior 7-128MB/AMDAthlon700
(6) Nimzo 8-128MB/AMDAthlon700
(7) Crafty 19.01-120MB/AMDAthlon700
(8) Comet B54-128MB/AMDAthlon700
(9) Patzer 3.11a-112MB/AMDAthlon700
(10) Little Goliath 2000v3.9-128MB/AMDAthlon700
10 programs
208 positions

Reading 'Offset.pgn'
8537 matches
Analysing database structure...
1 cluster(s)
```

```
Calculating ELO ratings... 15 iterations
CPU time = 1 sec
```

Im vorliegenden Fall wurden insgesamt 10 verschiedene Programme gefunden. Der Marathontest, der mit Fritz mitgeliefert wird und hier als Anwendungsbeispiel dient, umfasst 208 verschiedene Positionen (eigentlich 210, aber zwei davon sind doppelt vorhanden!).
Die Bedeutung der Datei »Offset.pgn« wird erst im zweiten Teil des Artikels erläutert. Insgesamt konnten 8537 Stellungsvergleiche zwischen den Programmen ausgewertet werden. Die Datenbank besteht aus einem einzigen zusammenhängenden Cluster (Näheres dazu findet sich in der dem Programm beiliegenden Readme-Datei). Es waren 15 Iterationen notwendig, um die Elorangliste zu berechnen. Die Rechenzeit betrug 1 Sekunde. Beenden Sie jetzt das Programmfenster durch einen weiteren Druck auf die *Enter*-Taste.
EloStatTS erzeugt insgesamt fünf verschiedene Dateien, in denen zahlreiche statistische Informationen über die Stellungen und die getesteten Programme abgelegt werden. Zum Abschluss des ersten Teils dieses Beitrags wollen wir uns zunächst die berechnete Elorangliste näher ansehen. Sie befindet sich in der Datei "rating.dat". Fritz speichert im Programmnamen neben der Engine standardmäßig auch noch die Größe der Hashtabellen und den Prozessortyp mit ab. In der folgenden Rangliste (Tabelle 4) wird aus Platzgründen lediglich der Enginename angegeben.
Die Reihenfolge der Programme richtet sich nach der erzielten Elozahl. Daneben werden noch angegeben: die 95-%-Fehlergrenzen (siehe CSS 1+2/2000), die Anzahl der ausgewerteten Stellungsvergleiche ("Matches"), der erzielte Score, die mittlere Elozahl der Gegner ("Average Opponent"), die Anzahl der gelösten Stellungen ("Solved Positions"), die mittlere Lösezeit für die gelösten Stellungen ("Mean Solution Time" Nr. 1, kurz MST1), die mittlere Lösezeit für alle Stellungen einschließlich der ungelösten (MST2) sowie ein so genannter "Rank Index", den wir im zweiten Teil des Artikels noch etwas genauer betrachten.

	Programm		Elo	+/-	Matches	Score	Av.Op.	S.Pos.	MST1	MST2	RIndex
1	Nimzo 8	:	2671	8	1753	61.2 %	2592	190/208	13.5s	38.3s	0.81
2	Hiarcs 8	:	2642	8	1733	56.6 %	2595	184/208	15.1s	47.9s	0.69
3	LG2000 v.3.9	:	2640	8	1746	56.4 %	2595	188/208	19.9s	46.6s	0.69
4	Fritz 8	:	2636	8	1716	55.7 %	2596	182/208	20.1s	55.0s	0.66
5	CTiger 14.0	:	2630	8	1707	54.7 %	2597	173/208	16.0s	63.6s	0.69
6	Junior 7	:	2606	8	1710	50.8 %	2600	175/208	25.6s	69.0s	0.60
7	Shredder 7	:	2592	8	1694	48.7 %	2601	174/208	23.9s	68.9s	0.54
8	Comet B54	:	2533	9	1679	39.4 %	2608	163/208	33.7s	91.1s	0.37
9	Crafty 19.01	:	2526	9	1671	38.2 %	2609	161/208	35.8s	95.2s	0.38
10	Patzer 3.11a	:	2516	9	1665	36.8 %	2610	154/208	36.8s	104.8s	0.35

Tabelle 4: Die mit EloStatTS erzeugte Rangliste (rating.dat)

Der Marathontest bewertet naturgemäß fast ausschließlich die taktischen Fähigkeiten der Programme, was in der Rangliste deutlich zu erkennen ist. Nimzo 8 schneidet mit Abstand am besten ab. Dahinter folgen die taktisch ebenfalls sehr starken Programme Hiarcs, Little Goliath und Fritz. Das Mittelfeld bilden Junior und Shredder. Comet, Crafty und Patzer liegen mit deutlichem Abstand am Ende der Liste.
Interessant ist ein Blick auf Lösequote und mittlere Lösezeit der Programme. Beide Größen werden hier lediglich aus Vergleichszwecken mit aufgeführt, da Sie die zentralen Größen aller bisherigen Stellungstests waren. Die mit EloStatTS berechnete Rangliste wird völlig unabhängig davon berechnet. Trotzdem existiert eine signifikante Korrelation zu diesen Größen. Im Einzelfall gibt es aber durchaus Abweichungen.
So hat z.B. Chess Tiger eine etwas schlechtere Lösequote als Junior und Shredder. Aufgrund seiner besseren mittleren Lösezeiten und seines deutlich höheren RIndex liegt er aber elomäßig deutlich vor beiden. Auch Hiarcs profitiert davon in seinem Vergleich mit Goliath, wo er sich trotz vier weniger gelöster Stellungen knapp behaupten kann. Es sollte an dieser Stelle noch einmal betont werden, dass diese Ergebnisse keine Widersprüche an sich darstellen. Sie resultieren vielmehr daraus, dass im Gegensatz zu den alten Berechnungsformeln keine Lösequote und keine mittlere Lösezeit zur Auswertung herangezogen wird, sondern stattdessen jeder einzelne Stellungsvergleich mit in das Endergebnis einfließt.
Im zweiten Teil des Artikels werden wir uns die anderen von EloStatTS erzeugten Dateien ansehen, die weiterführende Informationen über die getesteten Programme und die beteiligten Stellungen bereitstellen. Weiterhin werden wir auf einige Besonderheiten des Auswerteprogramms sowie auf die Testdurchführung unter Fritz im Allgemeinen eingehen. Bis dahin wünschen wir dem Leser viel Spaß beim Testen des Programms und beim Auswerten eigener Stellungstests. (*fs*)

Informationen zum Autor:

Dr. Frank Schubert

Download EloStat

File Title: EloStat *(Details)*
File Type: zip
File Version: 1.3
File Size: 73.27Kb