

DEEP BLUE

Hendrik Baier



Themen

- Matches Deep Blue – Kasparov 1996/97
- Faktoren von Deep Blues Erfolg
 - Systemarchitektur
 - Search Extensions
 - Evaluationsfunktion
 - Extended Book
- Vergleichstraining der Evaluationsfunktion



Teil eins

Beating the World Champion



Man vs. Machine

- Februar 1996 – Garry Kasparov vs. Deep Blue (I) 4 : 2
- Mai 1997 – Garry Kasparov vs. Deep Blue (II) 2,5 : 3,5
- Zweifel
- Ausblick



Kasparov

- Vor dem Match

"I made a firm decision for myself to take any challenge. And even if one day I'm doomed to lose, I thought it would be absolutely vital for us to play and to fight as long as we can."

- Nach dem Match



"I'm only a human being.
I get scared when I see something
beyond my comprehension."



Team Blue



Team Blue with Yasser and Maurice

Photo: Geoff McNiven

Teil zwei

Inside Deep Blue




Deep Blue

- Die Geschichte des Projekts
- Systemüberblick
- Der Chess Chip
- Die Hardware-Suche
- Die Software-Suche
- Parallelität
- Die Evaluationsfunktion
- Verschiedenes



Die Geschichte



System name	First played	Processors	Nodes/second	Feature groups
ChipTest	1986	1	50K	1
ChipTest-m	1987	1	400K	1
Deep Thought	1988	2–6	700K–2M	4
Deep Thought 2	1991	14–24	4M–7M	4
Deep Blue I	1996	216	50M–100M	32
Deep Blue II	1997	480	100M–200M	38
Deep Blue Jr.	1997	24	20M–30M	38
Deep Blue Jr. demo	1997	1	2M	38

Systemüberblick

- 
- 30-processor IBM RS/6000 SP Computer
 - 480 single-chip chess search engines (16 p.P.)
 - Nodes sind P2SC-Prozessoren 120 / 135 MHz mit je 1 GB RAM, 4 GB Disk
 - Betriebssystem: AIX 4.2
 - Drei-Schichten-Organisation
 - Grundideen
 - Systemgeschwindigkeit ~ 126 Mio. Pos./Sek.

Der Chess Chip

- Zuggenerierung
- Evaluationsfunktion
- Suchkontrolle
- Erweiterbarkeit



Hardware-Suche

- Null-Window Search
- 4 – 5-ply tief plus Quiescence
- binäre Suche möglich



Software-Suche

- hochselektives Suchverfahren:
„dual credit with delayed extensions“
- Mechanismen der Crediterzeugung
- Beispielverhalten

Iteration	Minimum software depth	Maximum software depth	Estimated maximum combined depth
6	2	5	11–21
7	3	6	12–22
8	4	11	17–27
9	5	15	21–31
10	6	17	23–33
11	7	20	26–36
12	8	23	29–39

Parallelität

- Algorithmen der parallelen Suche
- Implementierung der parallelen Suche
- Performance der parallelen Suche



Die Evaluationsfunktion

- Überblick

Function	Number of tables	Table entries	Data bits
Multiple pawns	2	80	8
Minor on weak	2	192	12
Self block	2	320	5
Opponent block	2	128	4
Back block	2	160	5
Stress	2	128	8
Mobility	8	128	9
Pawn structure	2	160	32
Passed pawns	2	320	32
Joint signature	1	256	8
Rooks on files	2	192	10
Bishops	4	128	11
Pawn storm	2	128	18
Pawn shelter	2	384	14
Development	1	256	9
Trapped bishop	1	128	8
Signature	2	128	20
Contempt	1	256	8
Piece placement	1	1024	10

- Beispiel: Türme auf Linien (Rooks on files)

- automatische Analyse der Evaluationsfunktion

Function	Number of registers	Data bits
Rooks on seventh rank	12	8
Bias	1	8
Opposing rook behind passed	1	9
Mpin and hung	1	7
Pinned and simple hung	1	8
Hung	1	7
Xraying	2	6
Pinned and hung	1	7
Permanent pin and simple hung	1	8
Knight trap	6	8
Rook trap	8	8
Queen trap	2	8
Wasted pawns	2	6
Bishop pair	2	7
Separated passed	2	8
Missing wing	2	10
Bishops of opposite colors	2	6
Evaluation control	2	32
Side to move	2	4

Verschiedenes

- Opening Book
- Extended Book
- Endspiel-Datenbank
- Zeitkontrolle



Teil drei

Tuning the Evaluation Function



Grundideen

- Methoden des Trainings Neuraler Netze:
 - Supervised Learning
 - Comparison Training
 - Reinforcement Learning
- Häufiges Problem: Overfitting
- Abhilfe: z.B. Early Stopping



Tesauros Arbeit

- Verallgemeinerung des Vergleichstrainings auf Suchtiefen > 1 ply
(angewendet auf das Training der Gewichte einer linearen Evaluationsfunktion)
- Experimente mit SCP
- Erfolgreiche Anwendung bei Deep Blue
- Ausblick

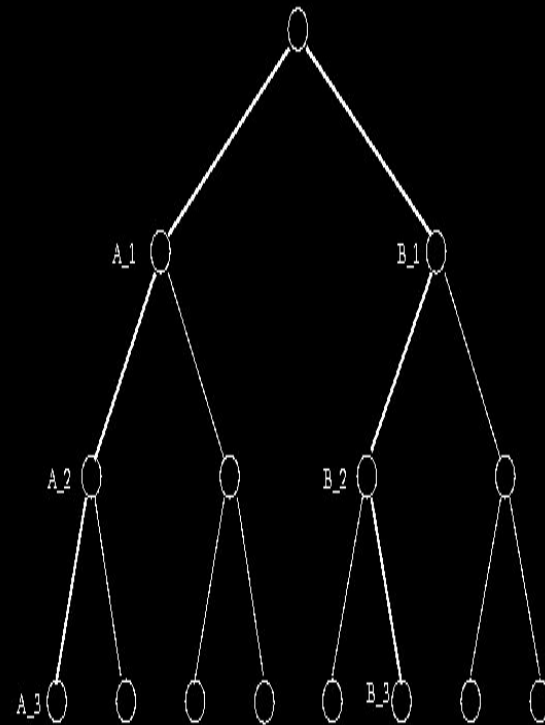


Problemstellung

- Schach benötigt einen Suchmechanismus, der viele Züge vorausrechnet
- Interaktion zwischen Suche und Evaluation ist nichttrivial
- Trainingssuchtiefe geringer als Performancesuchtiefe
- Zwei Arten von Verallgemeinerung erwartet




Verallgemeinerung



- Trainingsidee: multiple PVs
- Problem: keine Konvergenz?
- Lösungsidee: learning rate verringern, keine cutoffs

Experimente mit SCP

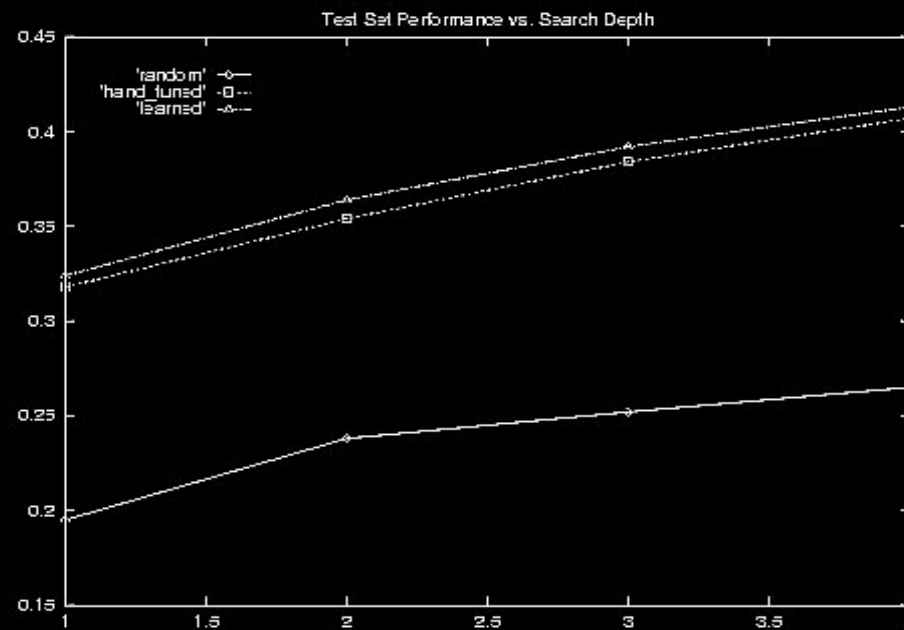
- Einfaches 1-ply-Training



Search Depth	Initial perf	Final perf
1-ply	.110	.207
1-ply + CQ	.293	.288
2-ply + CQ	.299	.288
3-ply + CQ	.343	.270

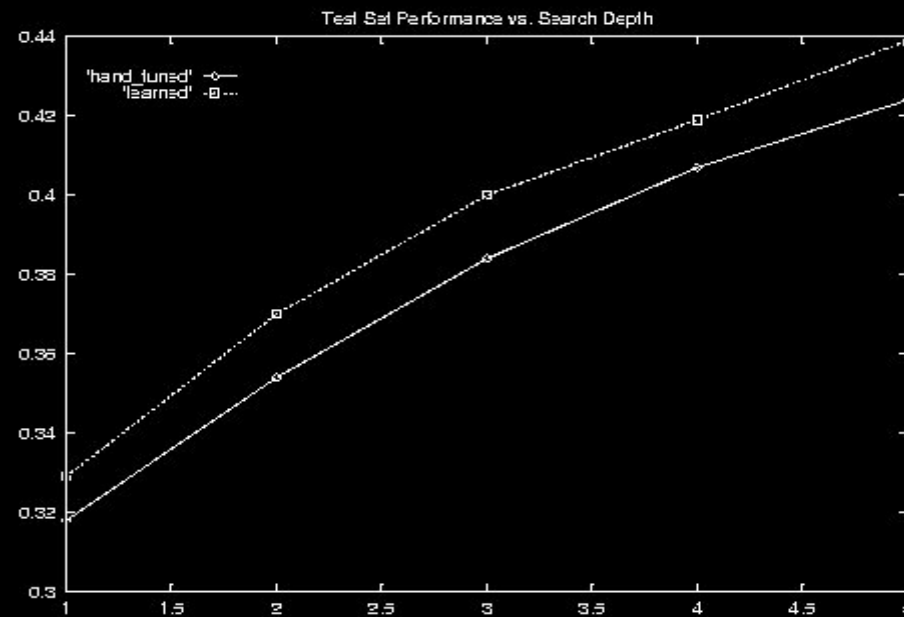
Experimente mit SCP

- 1-ply-Training mit Erweiterungen
mit zufälligen Gewichten gestartet



Experimente mit SCP

- 1-ply-Training mit Erweiterungen
mit voreingestellten Gewichten gestartet



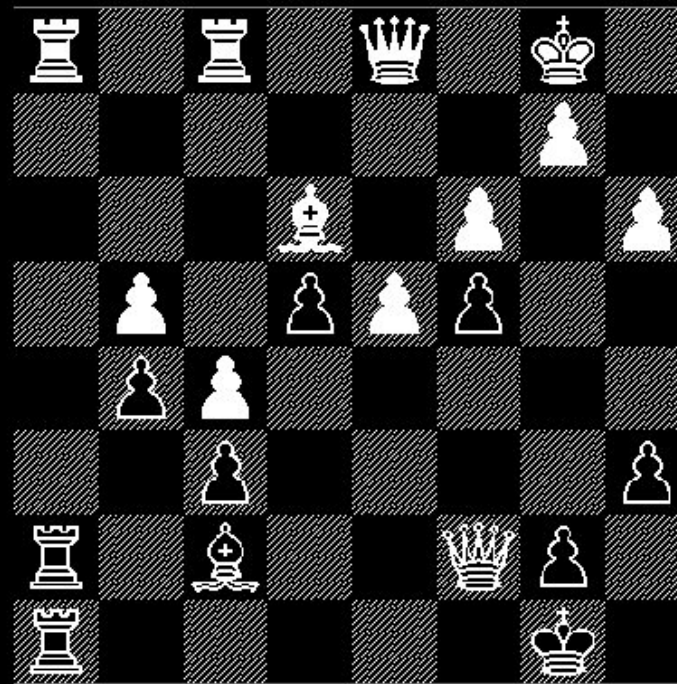
Anwendung bei Deep Blue

- Entscheidung für eine kleine Untermenge von Features (king safety)
- Anpassungen des Tuning-Algorithmus
- Auswirkungen auf das Match gegen Kasparov



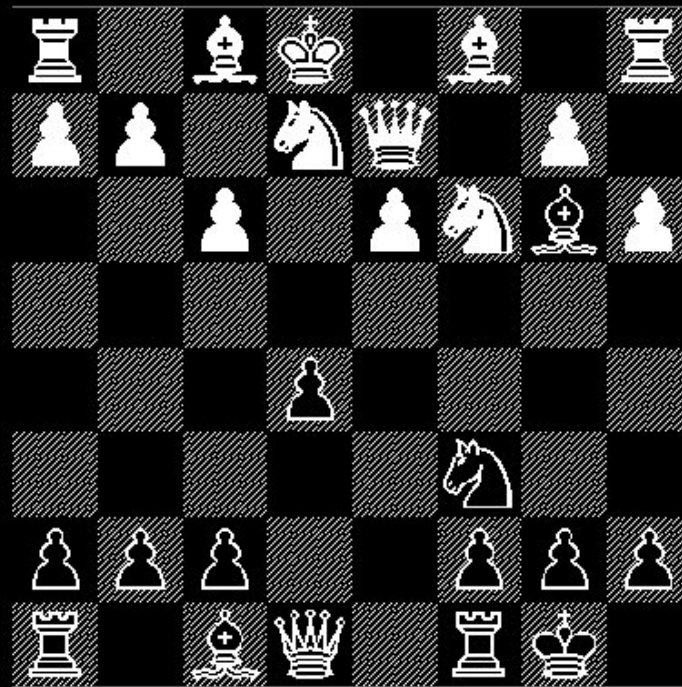
Auswirkungen

- Beispiel aus Spiel zwei



Auswirkungen

- Beispiel aus Spiel sechs



Ausblick

- GM-Imitation = gutes Spiel?
- andere algorithmische Formulierungen
- Othello, Go, Dame, Shogi
- multi-layer network evaluators (gewinnen an Bedeutung, sobald Suchstrategien keine Fortschritte mehr machen)



Quellen

- Murray Campbell, A. Joseph Hoane Jr., Feng-hsiung Hsu: Deep Blue. Volume 134, Number 1-2, January 2002, pp. 57-83.
- Gerald Tesauro: Comparison Training of Chess Evaluation Functions. In Fürnkranz & Kubat: Machines That Learn To Play Games, Nova Science Publishers 2001.
- Murray Campbell: Knowledge Discovery in Deep Blue. Communications of the ACM 42(11): 65-67, 1999.
- Google
- Google
- Google



