

Inhaltsverzeichnis

1	Einleitung	3
1.1	Computerschach und Künstliche Intelligenz	3
1.2	Geschichte des Computerschachs	3
1.3	Ergebnisse dieser Arbeit	6
1.4	Überblick über diese Arbeit	7
1.5	Grundlegende Definitionen	8
2	Der sequentielle $\alpha\beta$-Algorithmus	11
2.1	Die Scout-Variante	11
2.2	Heuristiken	19
2.2.1	Transpositionstabelle	20
2.2.2	Iteratives Vertiefen	26
2.2.3	Fenstersuche	27
2.2.4	Killerheuristik	27
2.2.5	Hauptvariante und Widerlegungsvarianten	28
2.2.6	Historyheuristik	29
2.2.7	Zugsortierung	29
2.3	Details der Implementierung	31
2.4	Hardware für den sequentiellen Algorithmus	33
2.5	Leistungsdaten des sequentiellen Algorithmus	34
2.5.1	Beschreibung des Testsets	34
2.5.2	Daten für verschiedene Suchtiefen	35
2.5.3	Scout vs $\alpha\beta$	35
2.5.4	Abstand vom minimalen Baum	35
2.5.5	Transpositionstabellennutzung	39
3	Der Parallele $\alpha\beta$-Algorithmus	41
3.1	Hardware für den parallelen Algorithmus	41
3.1.1	DeBruijn Netzwerk, Torus und Gitter im Vergleich	44
3.2	Bisher bekannte parallele Spielbaumsuchverfahren	44
3.2.1	Parallele statische Bewertung/Zugerzeugung	45
3.2.2	Parallele Fenstersuche	45
3.2.3	Spielbaumzerlegung	45
3.3	Das Konzept der dynamischen Spielbaumzerlegung	49
3.4	Grundversion des parallelen Scout-Algorithmus	50
3.4.1	Beginn einer Arbeitgeber-Arbeitnehmer Beziehung	50
3.4.2	Ende einer Arbeitgeber-Arbeitnehmer Beziehung	53
3.4.3	Fensterverbesserungen	53
3.4.4	Der verteilte Algorithmus	58
3.4.5	Gedanken zur Lastverteilung	65
3.5	Lokale Killer- und Historyheuristik	66
3.6	Die verteilte Transpositionstabelle	67
3.6.1	Lokale, globale oder verteilte Hashtabelle?	67

3.6.2	Implementierung einer verteilten Transpositionstabelle	69
3.7	Das Young Brothers Wait Concept	72
3.7.1	Das YBWC für Typ-1 und Typ-2 Knoten	74
3.7.2	Das verschärfte YBWC für Typ-2 Knoten	76
3.7.3	Kombination von YBWC-1-2 und YBWC*	77
3.8	Ausnutzung von Nachbarschaftsstrukturen	77
3.8.1	Notwendigkeit globaler Arbeitssuche	77
3.8.2	Lokale Arbeitssuche	79
3.8.3	Globale Arbeitssuche durch Rückmeldungen	80
3.8.4	Kombination lokaler und globaler Arbeitssuche	81
4	Verhalten auf bis zu 256 Prozessoren	83
4.1	Allgemeines	83
4.2	Streuungsmessungen	85
4.3	Verhalten der besten verteilten Version	90
4.3.1	Speedups für unterschiedliche Suchtiefen	90
4.3.2	Erklärung der gemessenen Speedups	91
4.3.3	"Superlinearer" Speedup in der Ruhesuche ?	95
4.3.4	Leistungsmessungen	96
4.3.5	Speedups unter Turnierbedingungen	98
4.3.6	Effizienz	101
4.3.7	Tiefe 8 Suche im DB(8): Einzelergebnisse	103
4.4	Leistungsverlustuntersuchungen	105
4.5	Kosten und Nutzen der verteilten Transpositionstabelle	107
4.5.1	Nutzen der verteilten Transpositionstabelle	107
4.5.2	Kosten der verteilten Transpositionstabelle	111
4.5.3	Fazit	113
4.6	Einschränkung des Suchoverheads durch das YBWC	114
4.7	Ausnutzung der Nachbarschaftsstrukturen	117
4.8	Ergebnisse der verschiedenen Topologien	125
4.9	Einfluß der dynamischen Lastverteilung	129
5	Verhalten auf mehr als 256 Prozessoren	131
5.1	Änderungen der Testumgebung	131
5.2	Die Standardversion	133
5.3	Scout vs $\alpha\beta$: Nullfenstersuche ?	135
5.4	Eingeschränkte Transpositionstabellennutzung	136
5.5	DB(8) vs. GC(32,32) in der Praxis	137
6	Zusammenfassung und offene Probleme	141
A	Eine BCH-Hashfunktion	143
B	Die Bratko Kopec Stellungen	147
	Literatur	149
	Index	153
	Abbildungsverzeichnis	157
	Tabellenverzeichnis	158