

Schach-Engine

Monte Carlo Tree Search

Alexander Hoerig & Luca Seufferle

Motivation/Inspiration

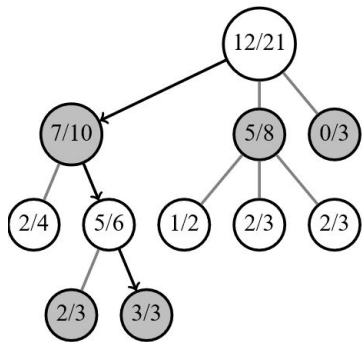


AlphaGo

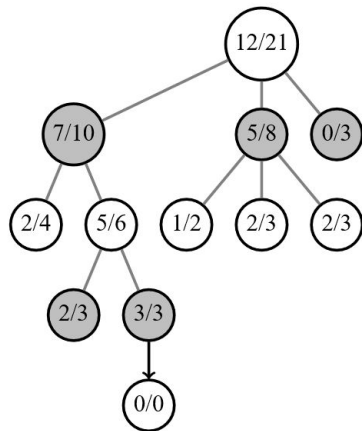
Verwendet Bewertungen von
Spielzügen,
Monte-Carlo-Methoden und
neuronale Netzwerke.

Algorithmus: Monte Carlo tree search

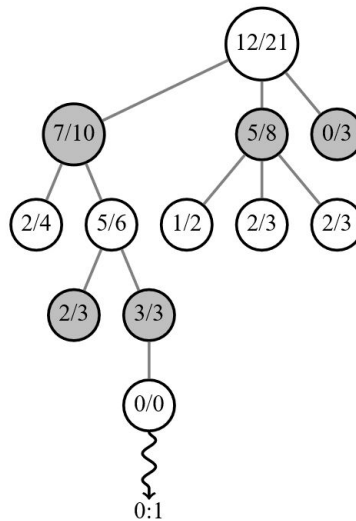
Selection



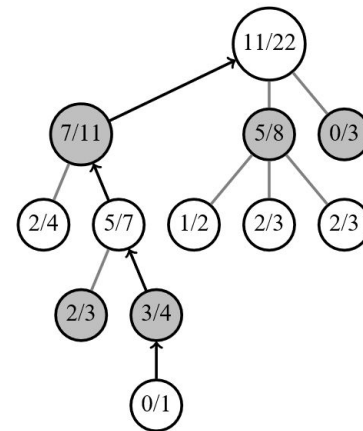
Expansion



Simulation

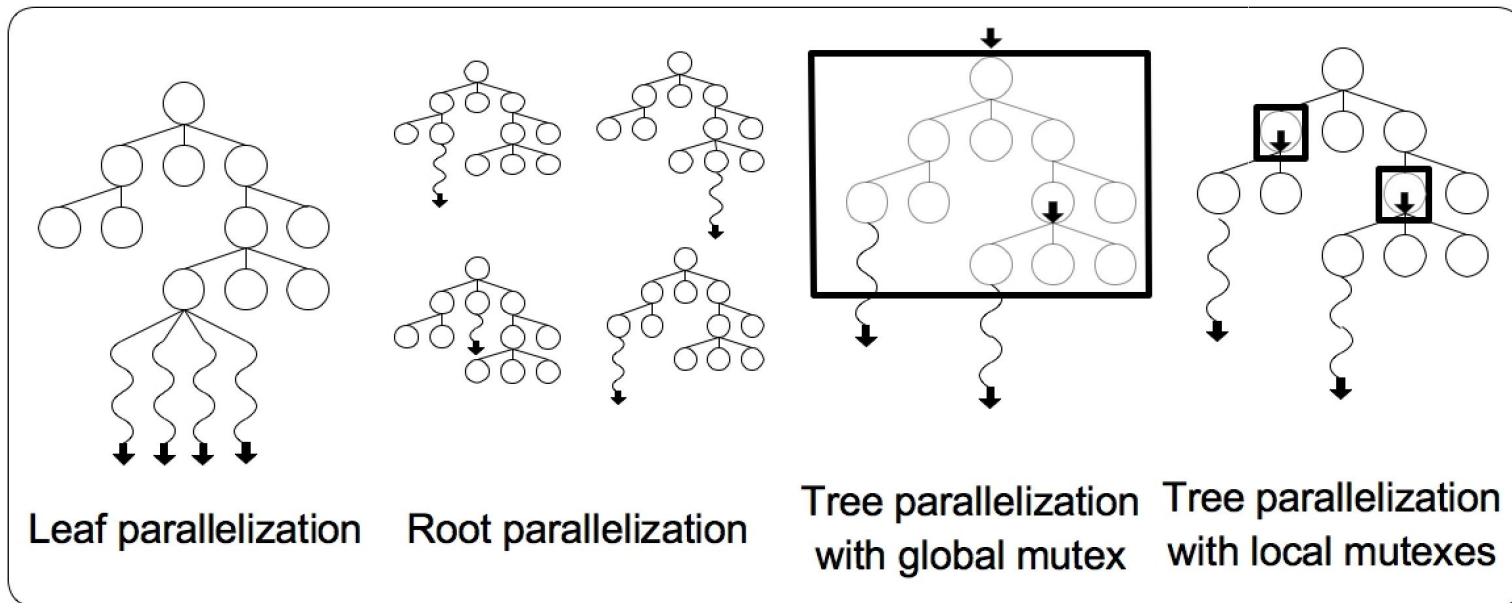


Backpropagation

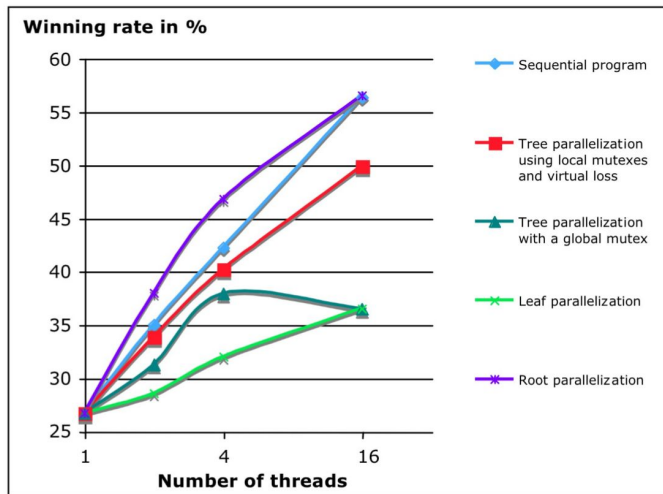
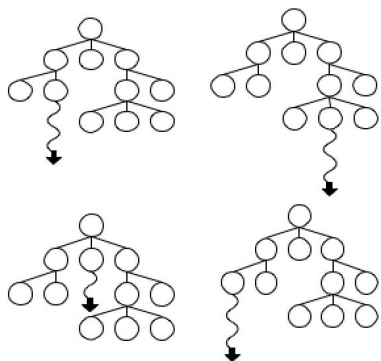


Parallelisierung

4 mögliche Ansätze:



Entscheidung: Root parallelization



- intuitiv
- Erwartung von guten Ergebnissen
- Wechsel auf MPI möglich (falls vielversprechend)
- gut über openMP Threads realisierbar

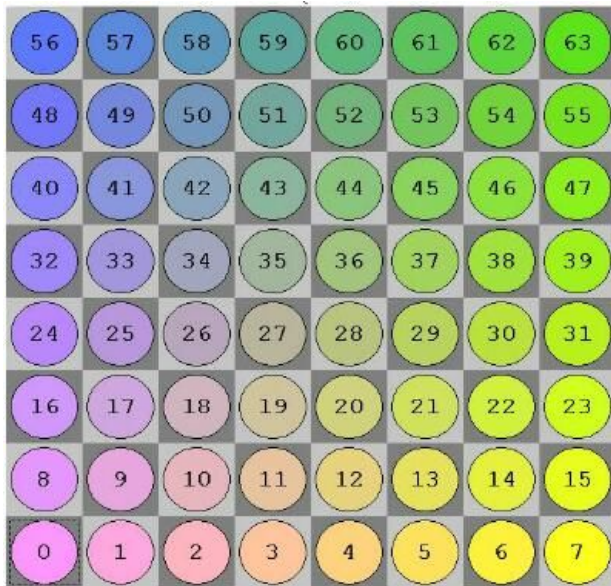
Brettdarstellung

pawns		knights	
white	black	white	black
.....
.....	. 1 . . 1 1 1
.....	1
.....	. 1 . . 1 1 . .
. . 1 1 . .
1 1 . 1 . 1 1 1 1 . .
..... 1 . .
..... 1
bishops		rooks	
white	black	white	black
.....	. 1	1 1 . .
..... 1
.....
. 1
. . 1	1 1
queens		kings	
white	black	white	black
.....	. 1 1
.....
.....
.....
.....
. 1

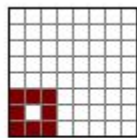
- Darstellung von Teilen des Schachbretts als Bitboards
- Einzig noch verwendete effiziente Darstellung
- Ermöglicht vertrautes Arbeiten für den Computer
- Benötigt höhere Abstraktionsfähigkeit



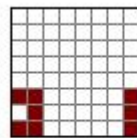
Verwendung von Bitboards



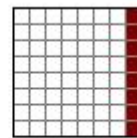
- 64-Bit unsigned Integer
- wichtige Eigenschaften:
 - erlaubt logikartige Verarbeitung
 - finden des kleinsten gesetzten Bits
 - Drehung und Spiegelung



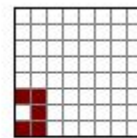
KING_ON_B2



SHIFTED_LEFT



FILE_H

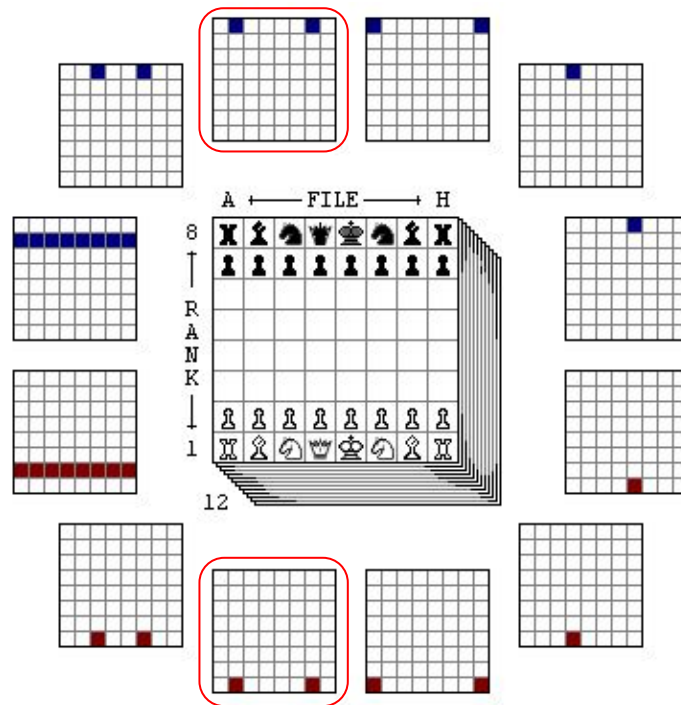


KING_ON_A2

Vektorisierung

- Dank Bitboards großes Potential
- potentielle Möglichkeiten:
 - bei der Zugerzeugung einer bestimmten Figur
 - beim Test auf Legalität
- anspruchsvolle Implementierung

In Binary (base 2)	01000010 00000000 00000000 00000000 00000000 00000000 00000000 01000010 <i>The squares B1, G1, B8, G8 are set to 1.</i>
In Hexadecimal (base 16)	42 00 00 00 00 00 00 42
In decimal (base 10)	4,755,801,206,503,243,842
In Pseudo-code	BITBOARD MySecondBitBoard := 4755801206503243842;



Probleme und Herausforderungen

— — —

Technisches:

- Bereits berechnete Teilbäume übernehmen
- Abstraktion von Bitboards

Sonstiges

- Kommunikation im Zweierteam ist nicht zu unterschätzen
- Einarbeiten neue Systeme (Git, Gtest, Deguggen, C++)

Ausblick

— — —

- Wechsel auf Tree parallelization und Root parallelization als MPI
- Optimierung der Zugerzeugung und des Legalitätstests
- Vektorisierung der Zugerzeugung und des Legalitätstests