MoGo

Seminar Knowledge Engineering und Lernern in Spielen
Sommersemester 2010

TECHNISCHE UNIVERSITÄT DARMSTADT

Inhalt

TECHNISCHE UNIVERSITÄT DARMSTADT

- Go
- Probleme für KIs

- Monte-Carlo-Suche
- UCT-Suchalgorithmus

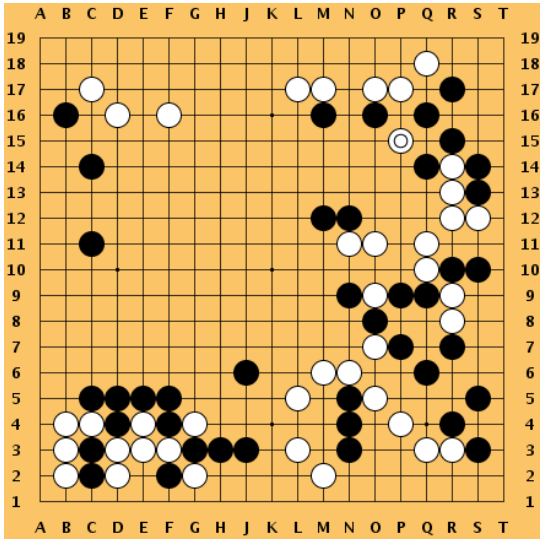
- Modifikationen in MoGo
  - Zugauswahl in der Monte-Carlo-Suche
  - Beschleunigte Zugbewertung
  - Einsatz von gelerntem Vorwissen

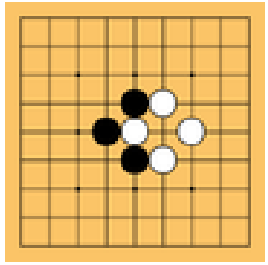
- Fazit
- Quellen und Abbildungsverzeichnis

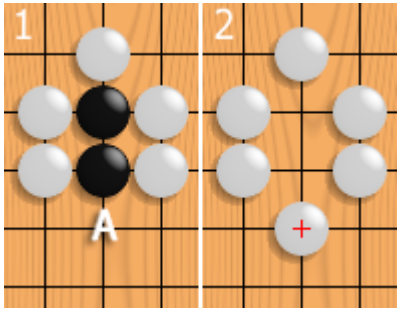
Go

TECHNISCHE UNIVERSITÄT DARMSTADT

- Brettspiel
  - 2 Spieler
  - 19x19 Felder (Anfänger: 5x5, 9x9)
  - Rundenorientiert
  - Schwarz beginnt
  - Starker Nachteil für weis
- Regeln
  - Schlagen möglich
  - Ko-Regeln gegen Wiederholungen
- Ziel
  - Schlagen von Steinen
  - Besetzen von Feldern







Probleme für KIs

TECHNISCHE UNIVERSITÄT DARMSTADT

- Extrem schwierig für KIs

- NP-vollständig
- Hoher Verzweigungsfaktor
  - Mögliche Züge < 361
- Stellungsvielfalt
  - $4{,}63 * 10^{170}$ (Schach: $\sim 10^{43}$)
- Keine geeignete Bewertungsfunktion oder Heuristik vorhanden
  - Nicht schnell genug
  - Nicht ausreichend gut

Monte-Carlo-Suche

TECHNISCHE UNIVERSITÄT DARMSTADT

- Ziel:
  - Bewertung von Nachfolgeaktionen einer Situation

- Simulation von Aktionsverläufen
- Bewertung durch Bewertung des Endzustandes
- Anschließend Berechnung des Mittelwertes aller Simulationen mit identischer Anfangsaktion

- Monte-Carlo-Suche approximiert die Bewertung einer Stellung

- Aktionswahl in der Simulation meist zufällig verteilt

UCT-Suche (1)

TECHNISCHE UNIVERSITÄT DARMSTADT

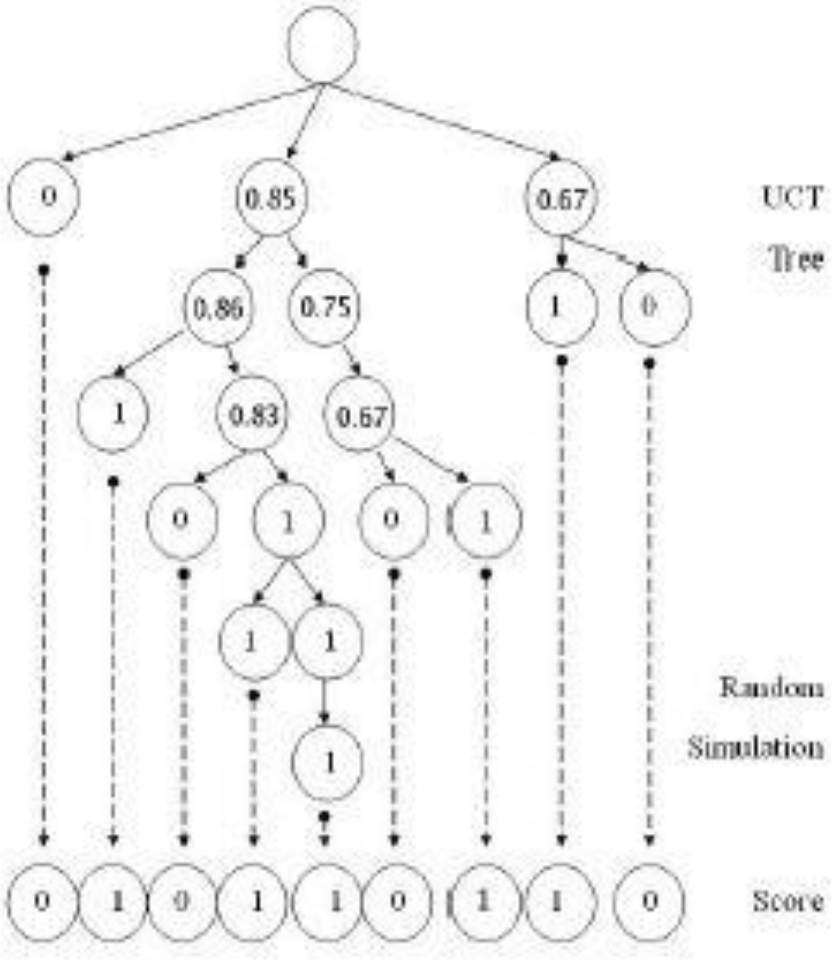
- **U**pper **C**onfidence bound applied to **T**ree

- Entscheidung für eine spezielle Aktion erfolgt aufgrund der aktuellen Nachfolgezustandsbewertung der möglichen Aktionen

- Approximation der Aktionsqualität mittels Simulationen
- Algorithmus mit 4 Phasen
  1. Selektion eines Spielzustandes
  2. Erzeugen von Nachfolgezuständen
  3. Simulation der Nachfolgezustände mit Monte-Carlo-Suche und Bewertung der Simulationen
  4. Anpassung der Spielzustandsbewertungen

$$\pi_{UCT}(s) = \arg\max_a \left[Q_{UCT}(s,a) + c\sqrt{\frac{\log n(s)}{n(s,a)}} \right]$$

UCT-Suche (2)


0
0.85
0.67
UCT
Tree
0.86
0.75
1
0
1
0.83
0.67
0
1
0
1
1
1
1
Random
Simulation
0
1
0
1
1
0
1
1
0
Score

Modifikationen in MoGo

TECHNISCHE UNIVERSITÄT DARMSTADT

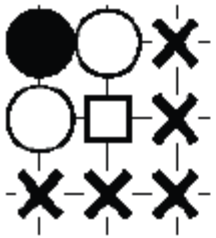
- Ziel:
  - Bewertung von Knoten beschleunigen
  - Möglichst kein Genauigkeitsverlust
- Zugbewertung mit Gewinnwahrscheinlichkeit
- Parallele Suche auf Baum
  - Ausschluss von Selektion des gleichen Blattes
- Zugauswahl in der Simulationsphase
- Beschleunigte Zugbewertung
- Einsatz von gelerntem Vorwissen

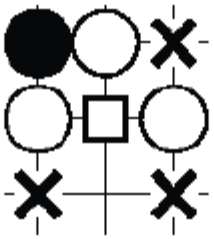
Zugauswahl in der Simulationsphase

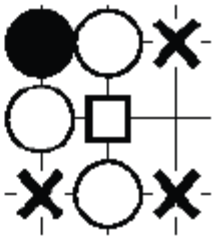
TECHNISCHE UNIVERSITÄT DARMSTADT

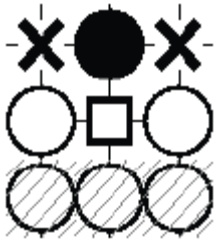
- Züge werden nicht mehr zufällig gewählt

1. beliebigen Zug zum sichern von Steinen
2. Suche eines Zuges mit Hilfe von Patterns im Bereich um den letzten Zug
3. Suche nach einem Zug der Steine gewinnt
4. Beliebiger Zug









Beschleunigte Zugbewertung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Hohe Verzweigung benötigt viele Simulationen
- Hohe Gefahr Züge falsch zu interpretieren
- Züge können verloren gehen

$$\pi_{UCT}(s) = \arg\max_a \left[Q_{UCT}(s,a) + c\sqrt{\frac{\log n(s)}{n(s,a)}} \right]$$

- Alt:
  - Bewertung des Anfangszuges einer Simulation
  - Durschnitt über alle Simulationen mit selben Anfangszug
- Neu:
  - Bewertung aller Züge einer Simulation
  - Durschnitt über alle Simulationen in denen der Zug vorkommt

$$\pi_{UR}(s) = \arg\max_a \left[\beta(s,a) \left[Q_{RAVE}(s,a) + c\sqrt{\frac{\log m(s)}{m(s,a)}} \right] + (1-\beta(s,a)) \left[Q_{UCT}(s,a) + c\sqrt{\frac{\log n(s)}{n(s,a)}} \right] \right]$$

Einsatz von gelerntem Vorwissen

TECHNISCHE UNIVERSITÄT DARMSTADT

- Initiale Bewertung oft ungenau
- Möglichst genau Initialbewertung eines Zuges wünschenswert

- Initialbewertung mittels gelernter Bewertungsfunktion
- Lineare Funktion von gewichteten binären Eigenschaften
  - Offline gelernt mit TD(0)-Algorithmus
  - Entnommen aus RLGO
  - Binäre Eigenschaften bestehen aus lokalen Steinkonfigurationen für 1x1 bis 3x3 große Feldbereiche
  - 2 Gewichte pro Steinkonfigurationen
  - Gespiegelte und rotierte Steinkonfigurationen teilen sich ein Gewicht
  - Steinkonfigurationen die zum selben Pattern gehören teilen sich das andere Gewicht

Fazit

TECHNISCHE UNIVERSITÄT DARMSTADT

- UCT ist gut geeignet für Suchprobleme mit hoher Verzweigung
- Go ist jedoch zu komplex für einen reinen UCT
- Mit modifizierter Approximation und ausreichend Rechenleistung vielversprechend

Algorithm	Wins .v. GnuGo
$UCT(\pi_{random})$	1.84 ± 0.22 %
$UCT(\pi_{MoGo})$	23.88 ± 0.85%
$UCT_{RAVE}(\pi_{MoGo})$	60.47 ± 0.79 %
$UCT_{RAVE}(\pi_{MoGo}, Q_{RLGO})$	69 ± 0.91 %

Fazit (2)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Titel von MoGo:
  - Sieger Computer Olympiade 2007
  - Zweiter Computer Olympiade 2008
  - Dritter Computer Olympiade 2009
  - 2009: Sieg über Zhou-Jun Xhou(9P) auf 9x9 Feld mit schwarz

Rank	Program	Country	Hardware	Score	Games	Title
1	Zen	JPN	2 x 2 core (4 x 2.8 GHz Opteron)	9.0	10	Gold medal
2	Fuego	CAN	8 x 8 core (64 x 3 GHz Xeon E5450)	8.0	10	Silver medal
3	MoGo	FRA	4 x 4 core (16 x 2.9 GHz Xeon)	7.0	10	Bronze medal
4	Fudo Go	JPN	6 PCs (20 x 3 GHz Core2 + 4 x 3.6 GHz Core i7)	3.0	10	
4	Go Intellect	USA	8 core (8 x 3 GHz)	3.0	10	
6	GoKnot-QYZ	ESP	4 core (4 x 3.6 GHz Core i7)	0.0	10	

Simulations	Wins .v. GnuGo	CGOS rating
3000	69%	1960
10000	82%	2110
70000	92%	2320*

		1	2	3	4	5	6
1	Zen		2	1	2	2	2
2	Fuego	0		2	2	2	2
3	MoGo	1	0		2	2	2
4	Go Intellect	0	0	0		1	2
5	Fudo Go	0	0	0	1		2
6	GoKnot-QYZ	0	0	0	0	0	

Quellen

- *Modification of UCT with Patterns in Monte-Carlo Go* , Gelly, S., Wang, Y., Munor, R. & Teytaud, O., INRIA (Techical Report 6062), 2006

- *Combining Online and Offline Knowledge in UCT*, Gelly, S. & Silver, D., Proceedings of the 24th International Conference on Machine Learning (ICML-07), Seite 279, 2007

Abbildungsverzeichnis

TECHNISCHE UNIVERSITÄT DARMSTADT

- Folie 1 und 3:
  - Quelle: http://en.wikipedia.org/wiki/Go_%28game%29
- Folie 7 und 9:
  - Quelle: *Modification of UCT with Patterns in Monte-Carlo Go* , Gelly, S., Wang, Y., Munor, R. & Teytaud, O., INRIA (Techical Report 6062), 2006
- Folie 12 und 13 links unten:
  - Quelle: *Combining Online and Offline Knowledge in UCT*, Gelly, S. & Silver, D., Proceedings of the 24th International Conference on Machine Learning (ICML-07), Seite 279, 2007
- Folie 13 Rest:
  - Quelle: http://www.grappa.univ-lille3.fr/icga/tournament.php?id=193