

Seminar aus maschinellem Lernen



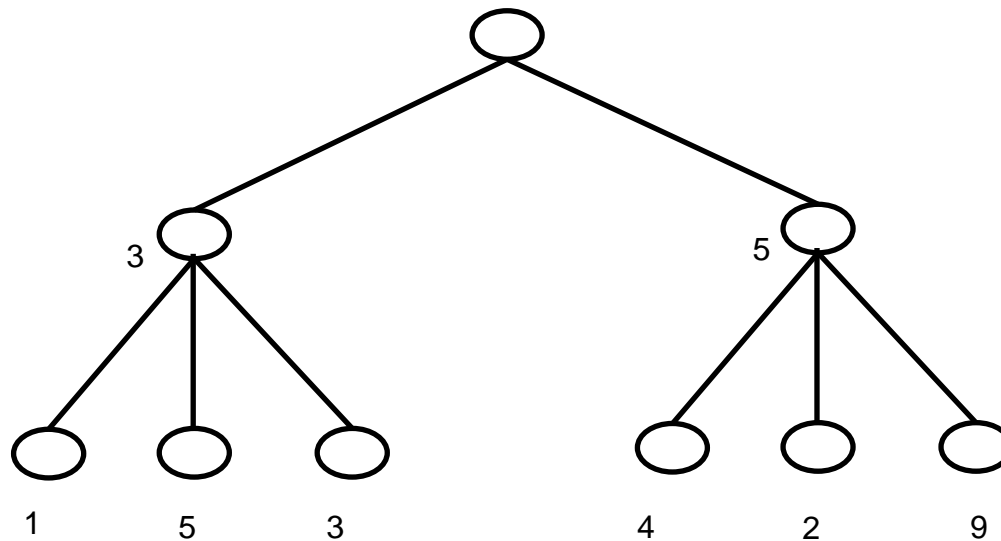
TECHNISCHE
UNIVERSITÄT
DARMSTADT

UCT vs. Minimax

- Algorithmen
- Motivation
- Baum Modell
- Noise Modell
- Fallen in Bäumen
- Zusammenfassung

Leaf Averaging

- Knotenwert ist Durchschnitt der Blätter



UCT (Upper Confidence Bounds for Trees)

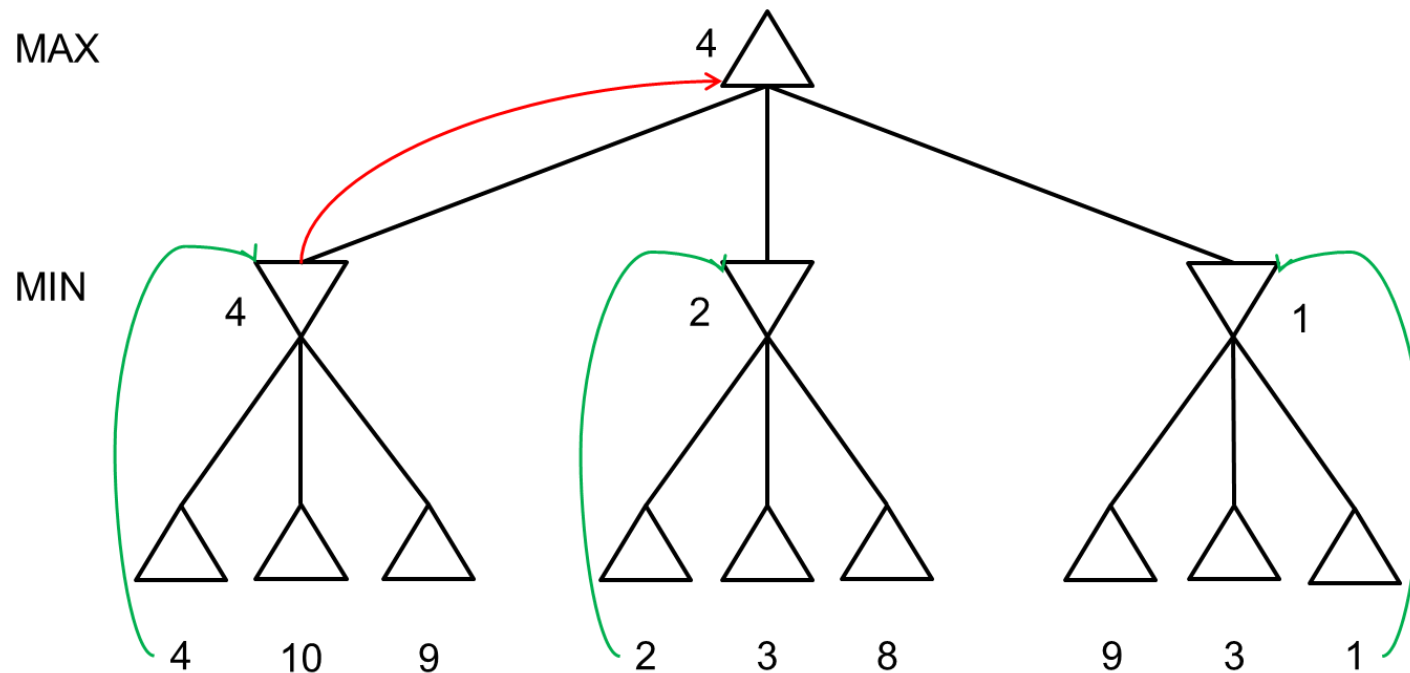
- Wählt Knoten für MCTS aus
- beschreibt Verhältnis von:
 - exploration: erkunde unbekannte Züge
 - exploitation: führe besten bekannten Zug aus

$$UCT = \bar{X}_j + 2C_p \sqrt{\frac{2 \ln(n)}{n_j}}$$

- konstante C_p beschreibt das Verhältnis exploration/ exploitation
- Wählt Nachfolgerknoten j aus der UCT maximiert
- Kerneigenschaft: Verhalten geht in Richtung Tiefensuche

Minimax

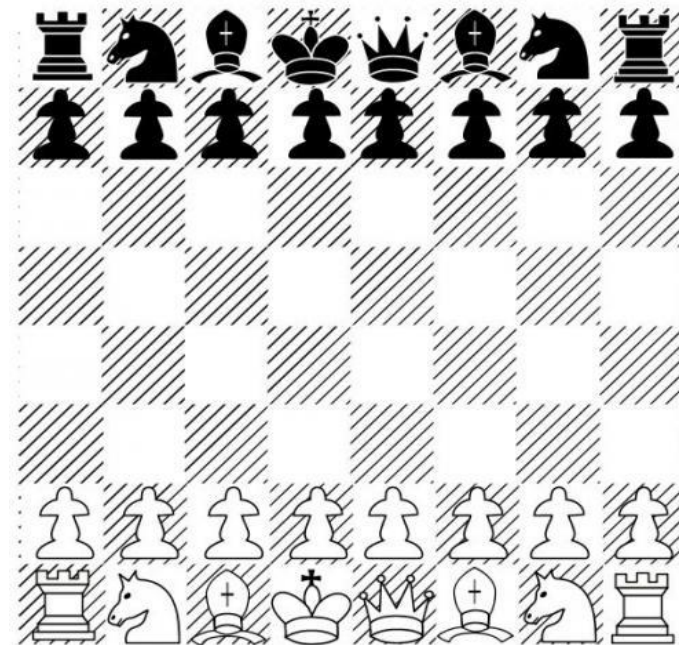
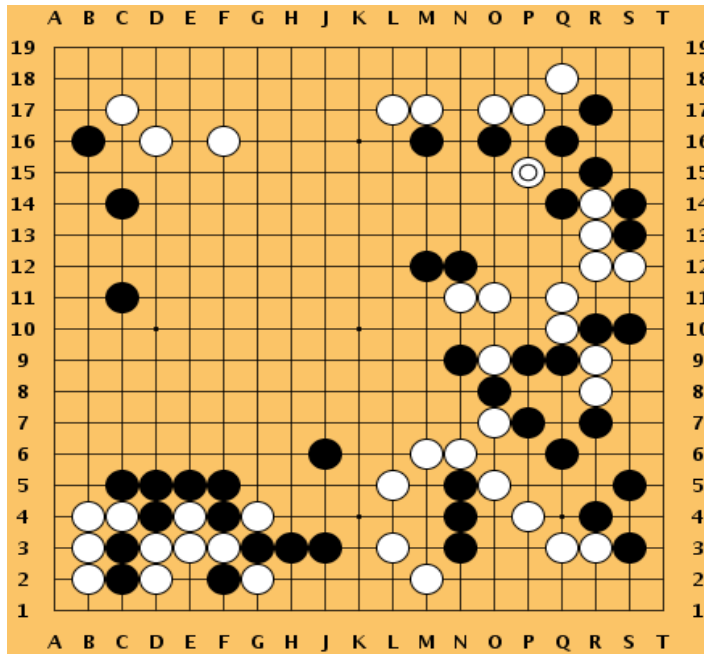
- MAX versucht den Wert zu maximieren, MIN zu minimieren



- Kerneigenschaft: Verhalten einer Breitensuche (bei Iterative Deepening)

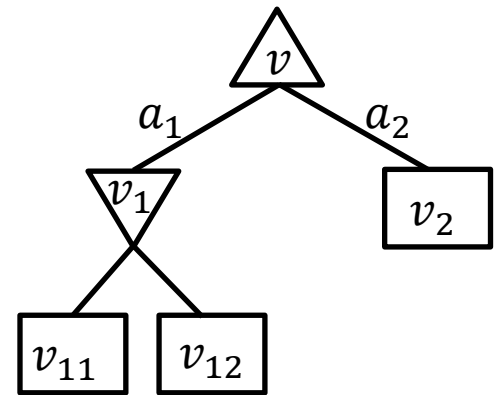
Motivation

Warum ist UCT in Go deutlich besser als Minimax, aber in Schach deutlich schlechter?



Baum Modell

- Baum: T
- Knoten: v
- Wert des Knotens : $m(v)$
- Nachfolger: $V = \{v_1, v_2, \dots, v_n\}$
- Aktionen in Knoten: $A = \{a_1, a_2, \dots, a_n\}$
- Grenzwert δ , wenn $|m(v)| > \delta \Rightarrow$ Blatt
- Wenn Aktion a optimal \Rightarrow Wert den Nachfolgers $m(v_i) = m(v)$,
sonst wenn Max am Zug $m(v_i) = m(v) - k$; für MIN $m(v_i) = m(v) + k$



Noise Modell: Independent Noise Modell

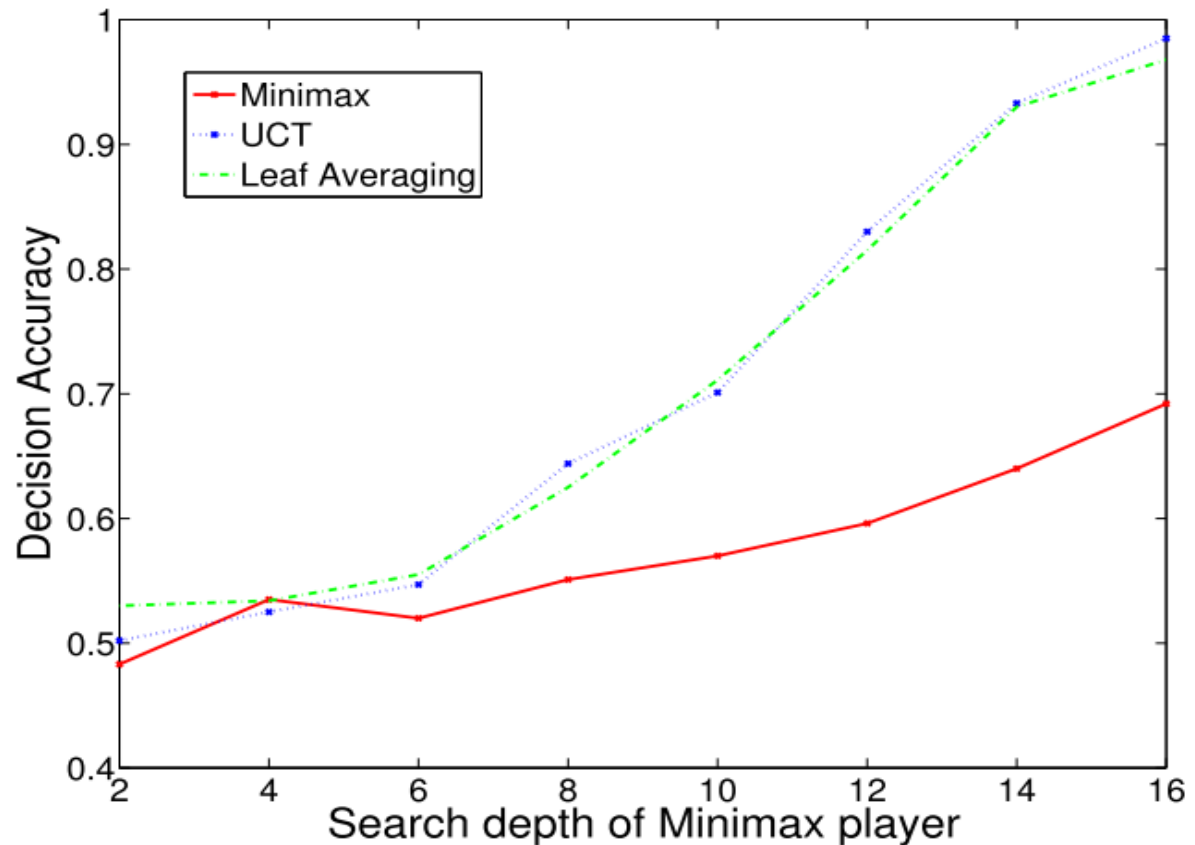
- Suchaufwand ist meist beschränkt:
 - Zeit
 - Tiefe
 - maximale Anzahl Knoten
 - ...
- Wenn Suche nicht in Blatt (Endzustand) endet, wird in realen Suchen eine Heuristik zur Evaluierung angewendet.
- Hier:

$$h(v) = m(v) + X$$

$$X \sim \mathcal{N}(0, \sigma)$$

Noise Modell: Independent Noise Modell

- gleicher Suchaufwand indem UCT und Leaf Averaging so viele Knoten expandieren dürfen wie Minimax Knoten besucht
- Kein rein akademisches Beispiel
- UCT deutlich besser



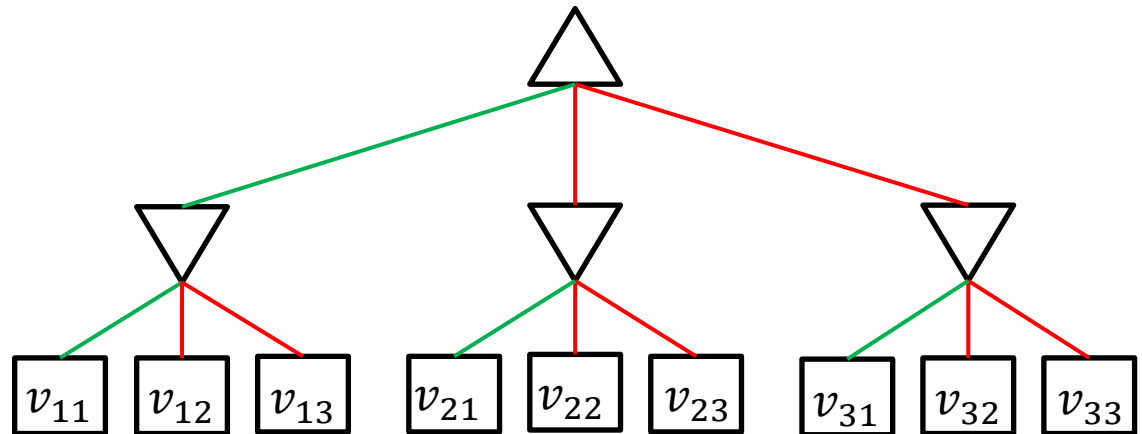
$$\sigma = 30; k = 1; \delta = d_{max} = \infty;$$

Noise Modell: Independent Noise Modell

- Leaf Averaging gibt Einblick warum UCT gut ist
- Man kann teilweise zeigen, dass der Wert von LA gegen $m(v)$ konvergiert
- Sei Y_v : *geschätzter Durchschnitt über alle Blätter*; Sei MAX am Zug
- Beweis von $\mathbb{E}[Y_v] = m(v)$ per Induktion:
- Induktionsanker: $d = 0$; $Y_v = m(v) + X \Rightarrow \mathbb{E}[Y_v] = m(v) + \mathbb{E}[N] = m(v)$
- Induktionsschritt: seien U die Nachfolger 2 Ebenen tiefer, b der Branching Faktor, b^2 Anzahl Nachfolger
- Sei Y_u *geschätzter Durchschnitt über alle Teilbaum Blätter für $u \in U$*
- Es folgt $Y_v = \frac{\sum_u Y_u}{b^2}$; $\mathbb{E}[Y_v] = \frac{\sum_u \mathbb{E}[Y_u]}{b^2}$; aus Hypothese: $\mathbb{E}[Y_v] = \frac{\sum_u m(u)}{b^2}$

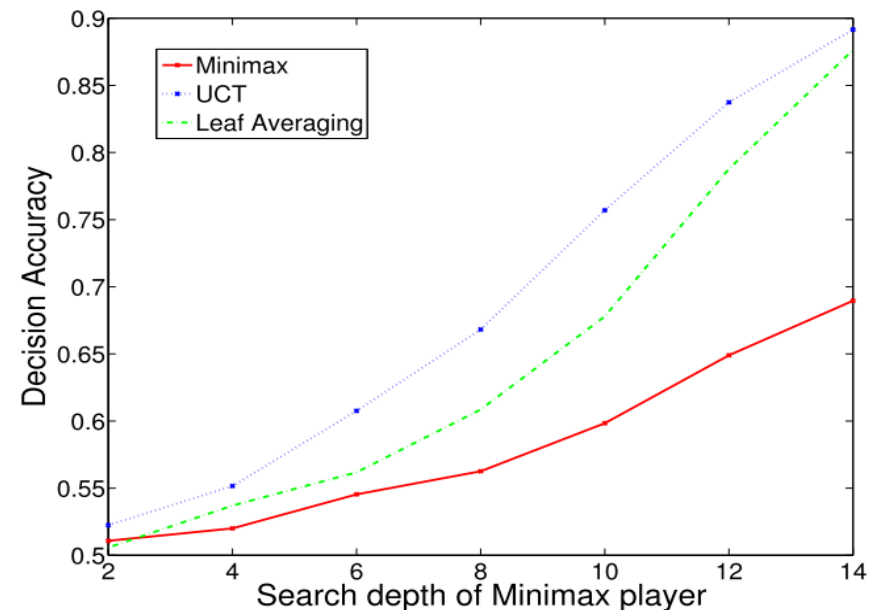
Noise Modell: Independent Noise Modell

- Aus Konstruktion des Baums:
 - $m(u) = m(v) + k, u \in \{v_{1,2}, v_{1,3}, \dots, v_{1,b}\}$
 - $m(u) = m(v) - k, u \in \{v_{2,1}, v_{3,1}, \dots, v_{b,1}\}$
 - Rest $m(u) = m(v)$
- $\Rightarrow \frac{\sum_u m(u)}{b^2} = m(v)$



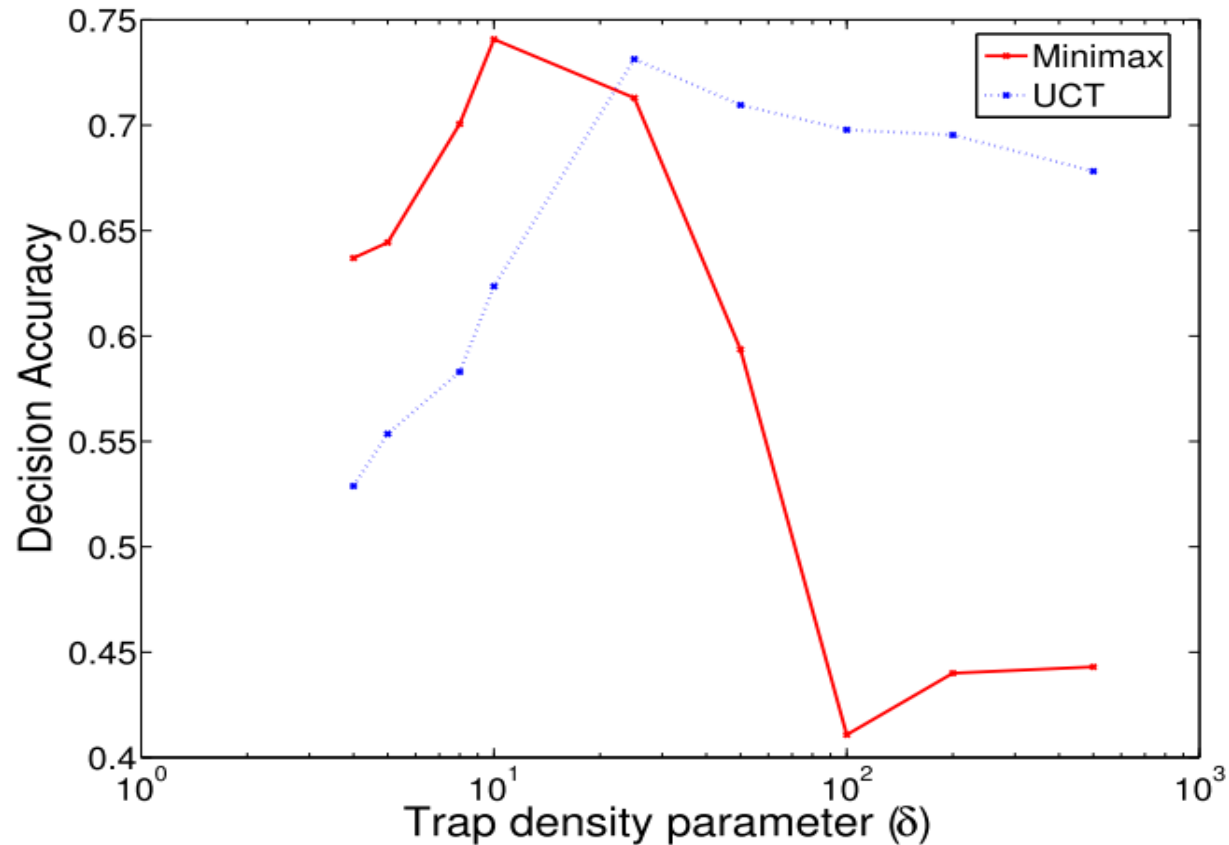
Noise Modell: Enhanced Independent Noise Modell

- Beobachtungen bei realen Spielsuchen:
 - Heuristik wird besser umso näher man dem Blatt ist
 - Manche Züge sind schlechter als andere
- Modellanpassungen:
 - $\sigma(v) = \min\{(\delta - |m(v)|), (d_{max} - d(v))\}$
 - zufälliges: $k \in \{1, \dots, k_{max}\}$



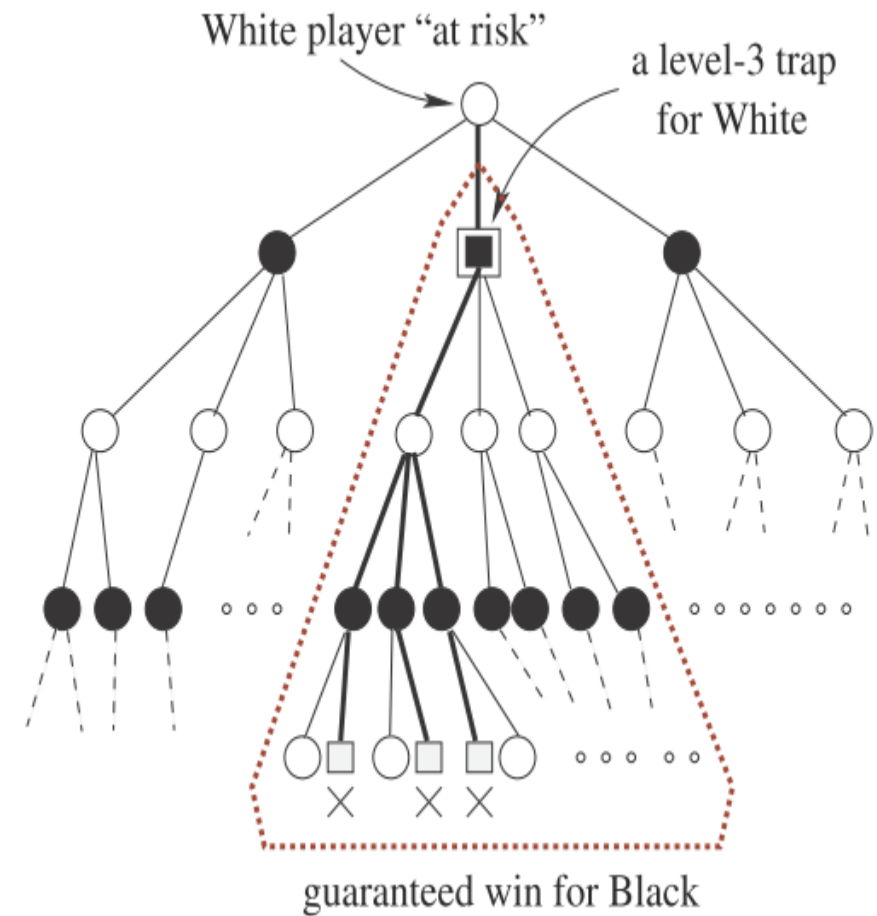
Noise Modell: Enhanced Independent Noise Modell

- Variation Anzahl der Endzustände/Fallen



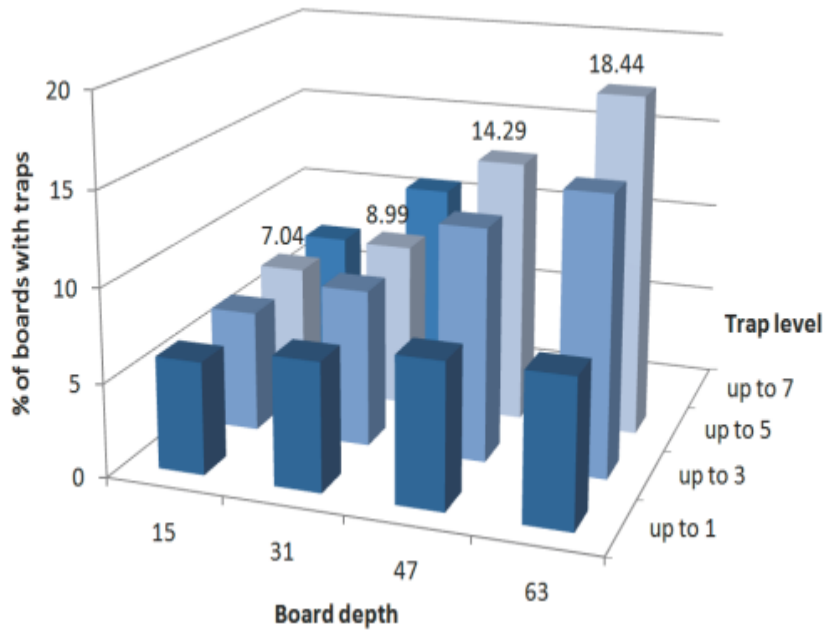
Fallen in Bäumen

- Jeder Zug führt zu Sieg, Niederlage oder Unentschieden
- Wenn Falle zu komplex/ tief für Berechnung ist wird diese nicht gezählt
- Fallen können mit iterativ Deepening am gut entdeckt werden

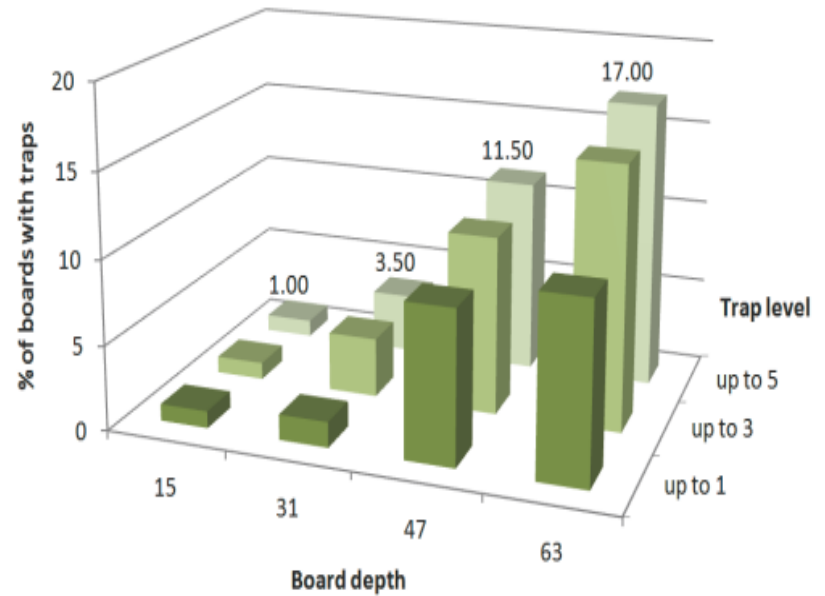


Fallen in Bäumen

- Kommen Fallen in realen Spielen vor?



200 Zufällig generierte Spielbretter



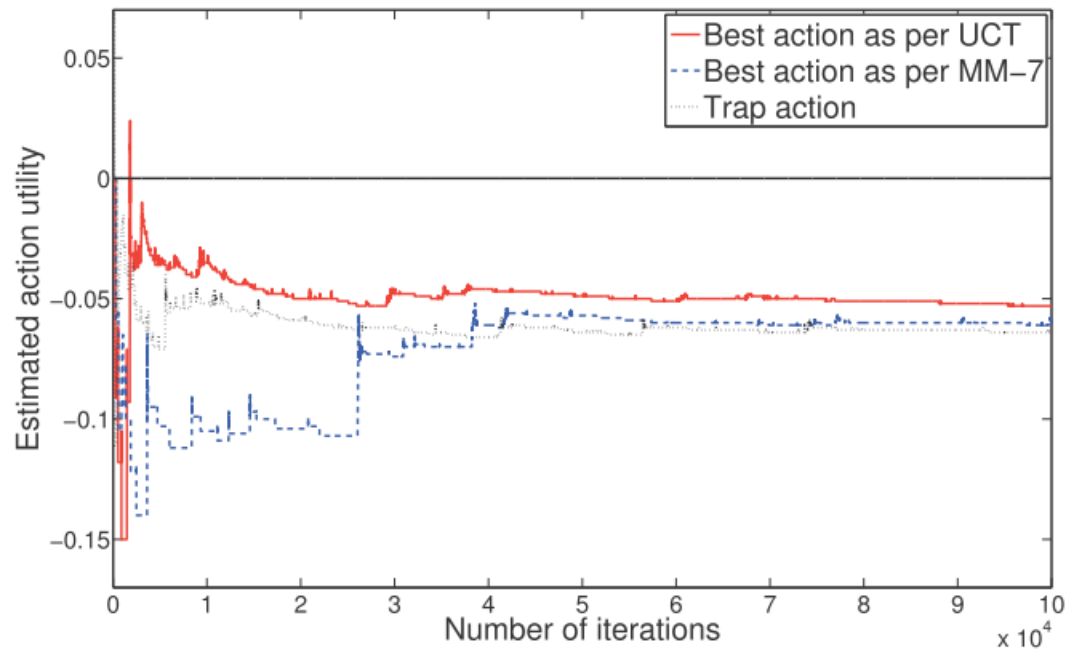
200 Spielbretter aus Profipartien

Fallen in Bäumen: UCT vs. Minimax

- Minimax entdeckt k-level Falle in einer Suchtiefe von $k+1$
- UCT darf so viele Knoten expandieren wie Minimax besucht hat
- Utility Werte im Bereich $[-1, +1]$
 - -1 : garantierter Loss
 - $+1$: garantierter Win

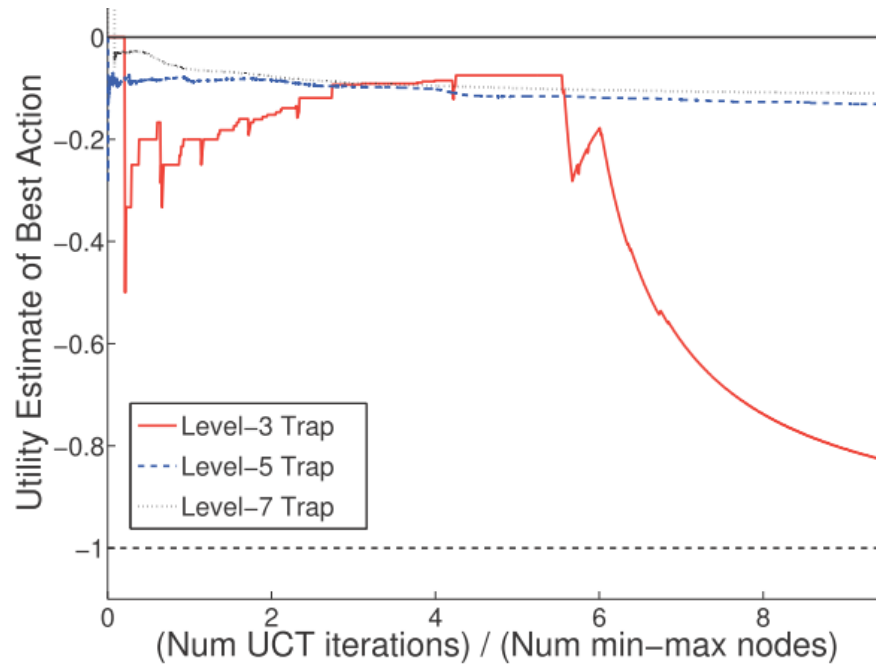
Fallen in Bäumen: UCT vs. Minimax

	UCT-best	Minimax-best	Trap move
level-1 trap	-0.083	-0.092	-0.250
level-3 trap	+0.020	+0.013	-0.012
level-5 trap	-0.056	-0.063	-0.066
level-7 trap	+0.011	+0.009	+0.004

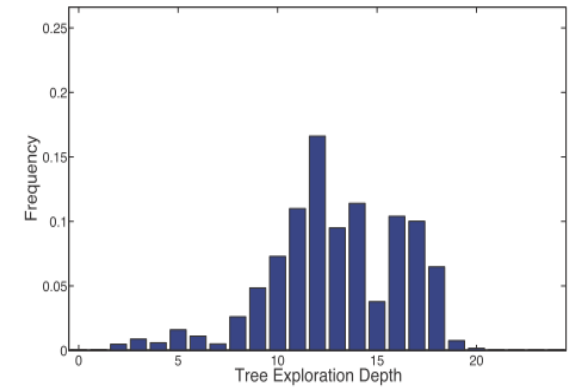
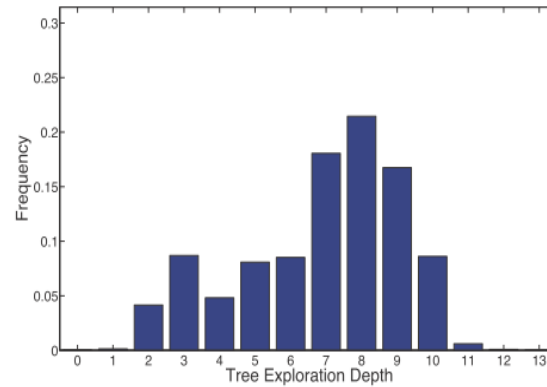
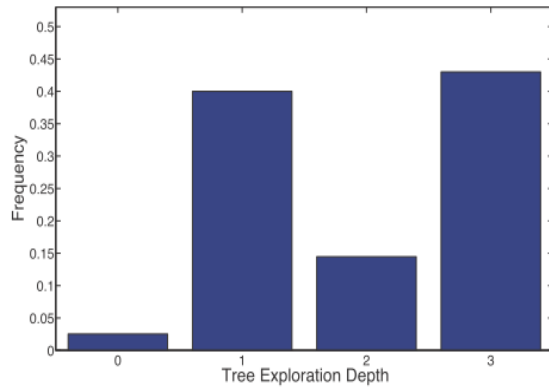


Fallen in Bäumen: UCT vs. Minimax

- Bis zu 10 mal mehr Aufwand für UCT
- UCT startet in Falle, damit dieser Knoten immer besucht wird



Fallen in Bäumen: UCT vs. Minimax



- Suchttiefen UCT für Fallen auf Level 3,5 und 7
- Level 3: 45% Aufwand für Level 3
- Level 5: 70% Aufwand tiefer als Level 5
- Level 7: 95% Aufwand tiefer als Level 7

- Minimax schwach bei hohem Branching Factor und schlechter Heuristik
- Die Performance von UCT hängt von der Art der Struktur des Suchraums ab.
- UCT hat Schwierigkeiten in Suchräumen mit Fallen
 - z.B.: Schach
- UCT ist Stark in harmonischen Suchräumen
 - z.B.: Go



Danke für die Aufmerksamkeit.
Gibt es noch Fragen?