

Improving heuristic mini-max search by supervised learning

Autor des Papers: Michael Buro

Evaluation: GLEM

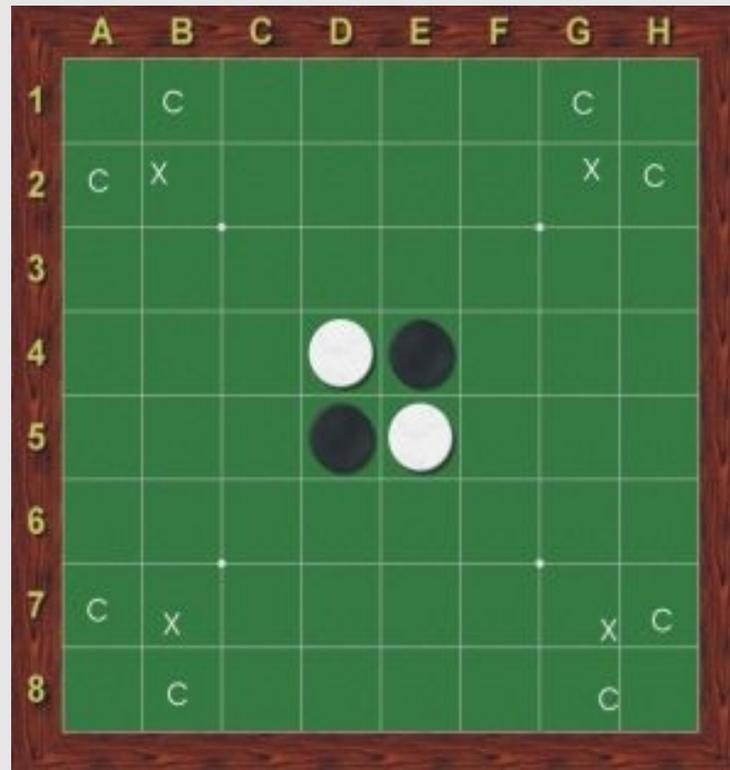
ProbCut

- Multi-ProbCut
- EndCut

Opening book construction

Logistello

Programm für Othello (Reversi)



Evaluation: GLEM

ProbCut

- Multi-ProbCut
- EndCut

Opening book construction

GLEM

„generalized linear evaluation model“

$$e(p) = g\left(\sum w_i * c_i(p)\right)$$

c: Konfiguration

w: Gewichtung

g: $\mathbb{R} \rightarrow \mathbb{R}$: wachsend und ableitbar

GLEM: Konfigurationen

$c = r_1(p) \& \dots \& r_n(p)$

$\text{val}(c(p)) = 1$ wenn belegt, 0 sonst

Positionen belegt oder nicht belegt, tritt also ein oder nicht

GLEM: Konfigurationen generieren

Function GenConf

Input: atomic feature set A , training position set E , minimal match count n

Output: configurations over A that are active in at least n positions of E

$R := \{\{f(\cdot) = k\} \mid f \in A, k \in \text{range}(f), \#\text{match}(\{f(\cdot) = k\}, E) \geq n\}$

$C := R$; collects all valid configurations

$N := R$; set of configurations created in previous iteration

while $N \neq \emptyset$ **do**

$M := \emptyset$; set of valid configurations in current iteration

(*) **foreach** $c \in N, d \in R$ **do**

$e := c \cup \{d\}$; specialize configuration c

if $\#\text{match}(e, E) \geq n$ **then**

$M := M \cup \{e\}$; append if valid

endif

endfor

$N := M$; next configurations to specialize

$C := C \cup N$; add valid configurations

endwhile

return C

GLEM (Forts.)

$$E(w) := \sum (s_i - e_w(p_i))^2$$

Dieser Fehler soll minimiert werden,
Gewichtungen entsprechend wählen

s_i : Positionswert

GLEM: Gewichte berechnen

Auch Gewichte kann man berechnen lassen

$$\delta^{(t)} = -\alpha \cdot (\text{grad}_{\mathbf{w}} E)(\mathbf{w}^{(t)})$$
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \delta^{(t)}.$$

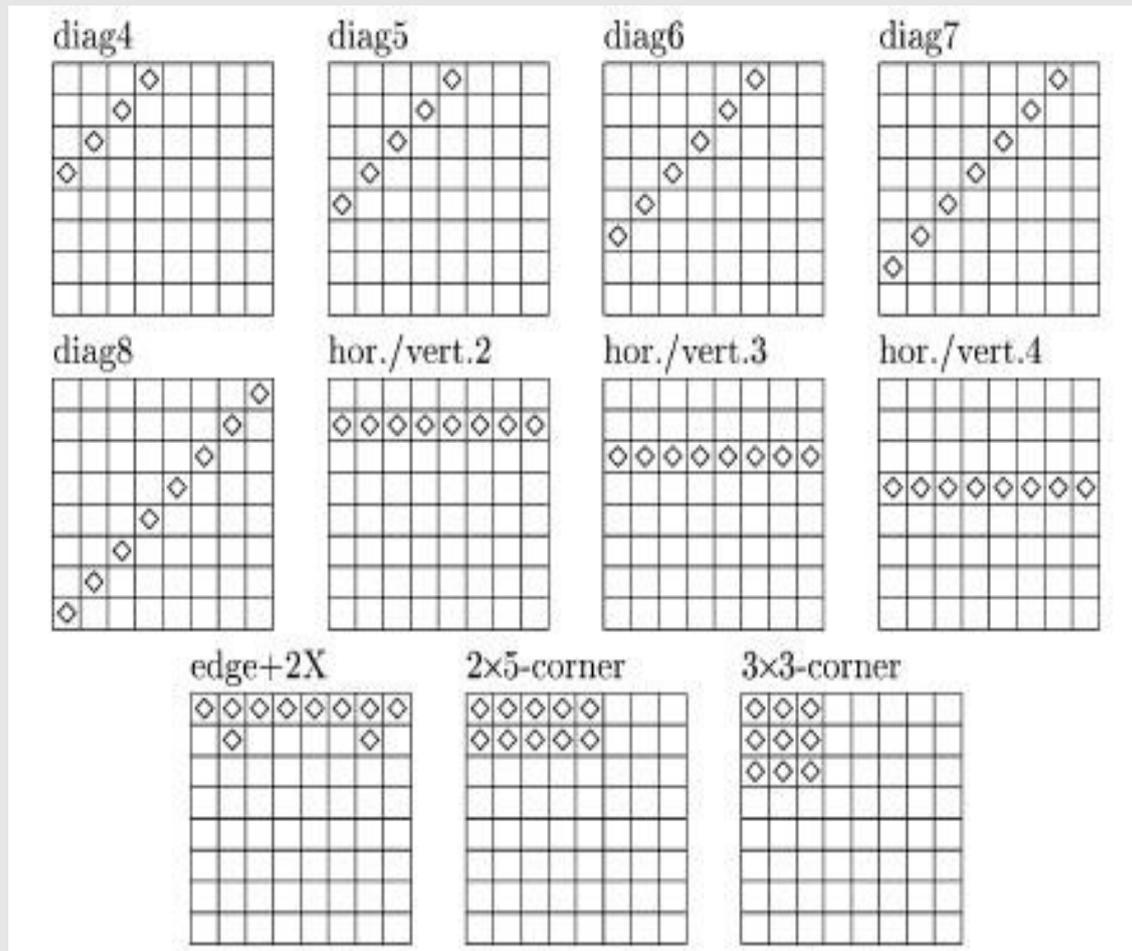
alpha: stepsize

grad_w E ist ein Vektor, die partiellen

Ableitungen:

$$\frac{\partial E}{\partial w_i}(\mathbf{w}) = -\frac{2}{N} \sum_{k=1}^N \Delta_k(\mathbf{w}) h_{i,k}.$$

Beispiele für Patterns



Beispiel Logistello

$$\begin{aligned} f(p) = & (\\ & [f_{d4,s.1} + \dots + f_{d4,s.4}] + [f_{d5,s.1} + \dots + f_{d5,s.4}] + \\ & [f_{d6,s.1} + \dots + f_{d6,s.4}] + [f_{d7,s.1} + \dots + f_{d7,s.4}] + \\ & [f_{d8,s.1} + f_{d8,s.2}] + [f_{hv2,s.1} + \dots + f_{hv2,s.4}] + \\ & [f_{hv3,s.1} + \dots + f_{hv3,s.4}] + [f_{hv4,s.1} + \dots + f_{hv4,s.4}] + \\ & [f_{edge+2X,s.1} + \dots + f_{edge+2X,s.4}] + \\ & [f_{2 \times 5,s.1} + \dots + f_{2 \times 5,s.8}] + \\ & [f_{3 \times 3,s.1} + \dots + f_{3 \times 3,s.4}] + f_{parity,s}) (p), \end{aligned}$$

Effekt von GLEM

| d | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-----|----|----|----|----|----|----|----|----|----|
| +0 | 34 | 32 | 32 | 35 | 34 | 31 | 26 | 26 | 32 |
| +1 | 59 | 56 | 46 | 57 | 50 | 41 | 44 | 40 | 38 |
| +2 | 83 | 75 | 70 | 62 | 61 | 59 | 48 | 51 | 54 |

GLEM Zusammenfassung

GLEM kombiniert also eine Sammlung von Features, die gewählt wurden

Zusammengefasste Features = Patterns

Da Konfigurationen 1 oder 0 werden, entscheiden die Gewichte alleine über ihre Wichtigkeit

Evaluation: GLEM

ProbCut

- Multi-ProbCut
- EndCut

Opening book construction

ProbCut

Schneide Teilbäume ab, die wahrscheinlich min-max-Wert nicht beeinflussen: forward pruning.

1. Shallowsearch liefert v_s .

2. Wenn

$$a * v_s + b$$

außerhalb α/β , also außerhalb von

$$[\alpha - t * \sigma, \beta + t * \sigma]$$

Suche abbrechen

3. Ansonsten: Wahren Wert v_d berechnen

ProbCut: Unterschied zu α/β -Pruning

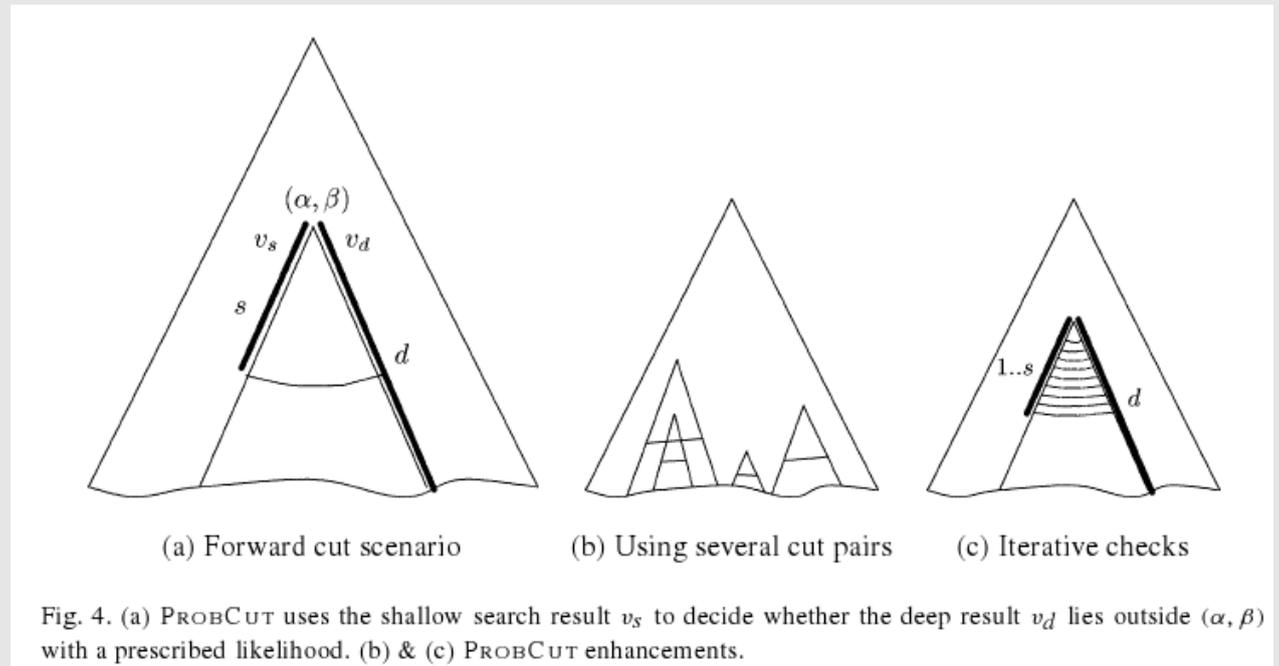
α/β -Pruning-Prinzip: Subtrees ignorieren, die den min-max-Wert nicht beeinflussen werden.

Aber: Immer noch muss auf unterster Ebene evaluiert werden

prune backwards

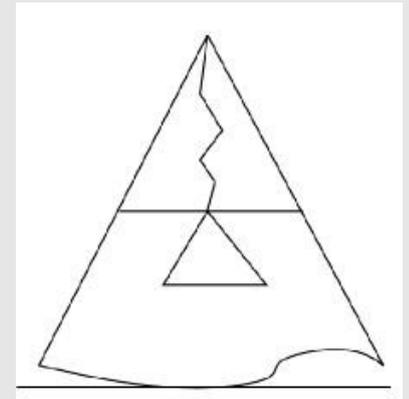
Multi-ProbCut (MPC)

- Parameter: Je nach Spielsituation unterschiedlich tief suchen
- Suchen mehrmals mit zunehmender Länge durchführen



EndCut

Ab bestimmter Tiefe: Schätze zuerst mit Shallowsearch, weiter werdend, wie das Spiel ausgeht.



ProbCut Anwendungen

MPC wurde inzwischen auch für Schach getestet ²:

„CRAFTY’s speed chess tournament score went up from 51% to 56%.“

Evaluation: GLEM

ProbCut

- Multi-ProbCut
- EndCut

Opening book construction

Opening Book Construction

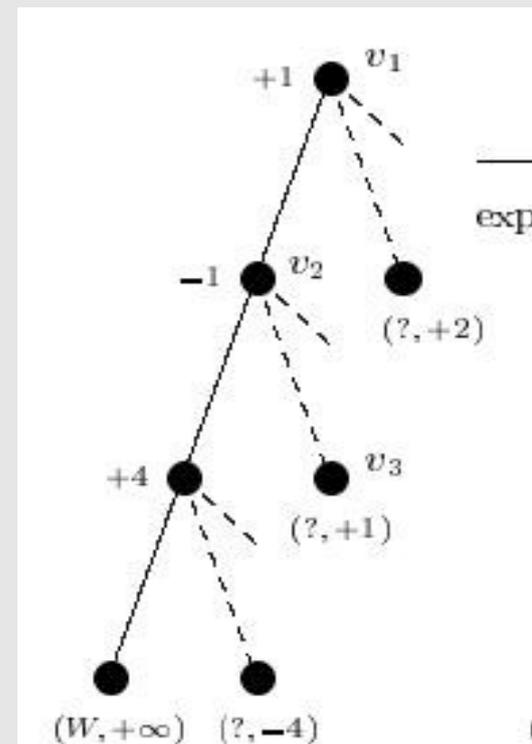
- Verhindere, wiederholt zu verlieren
- z.B. wichtig, wenn das Programm als Server läuft

Opening Book Construction: Der Baum

Spielbaum der Positionsvariationen, mit
((W, L, D, ?), $[-\infty, +\infty]$)

Für Win, Loss, Draw und unbekannt und den
Wert (dann geschätzt).

Außerdem: Heuristisch
beste Abweichung



Opening Book Construction: Vorgehen

„Find the node corresponding to the current position, propagate the heuristic evaluations from the leaves to that node by means of the nega-max algorithm, and choose the move that leads to the successor position with lowest evaluation.“

Opening Book Construction: Vorgehen (Bild)

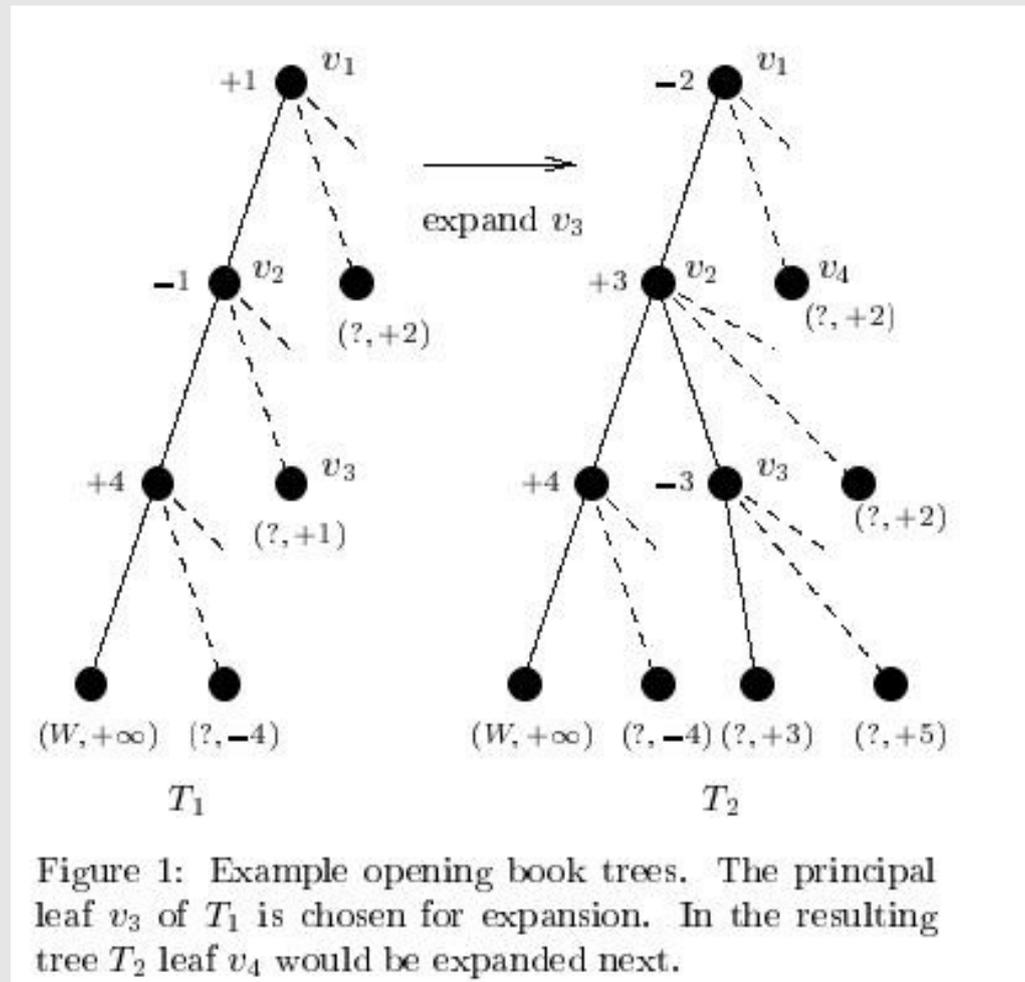


Figure 1: Example opening book trees. The principal leaf v_3 of T_1 is chosen for expansion. In the resulting tree T_2 leaf v_4 would be expanded next.

Fragen?

Quellen

Slide 2, Bild: Wikipedia, http://upload.wikimedia.org/wikipedia/de/6/6b/Othello_start.jpg

Genconf: Aus M. Buro: „From Simple Features to Sophisticated Evaluation Functions“

[1]: Aus M. Buro: „From Simple Features to Sophisticated Evaluation Functions“

Endcut: Aus Präsentation „Multi-ProbCut Search“,

<http://webdocs.cs.ualberta.ca/~mburo/ps/mpc.pdf>

[2] A.X. Jiang, M. Buro, „FIRST EXPERIMENTAL RESULTS OF PROBCUT APPLIED TO CHESS“

Opening Book Construction: M. Buro, „Toward Opening Book Learning“