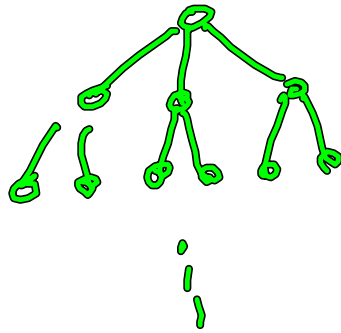


Suchraumproblematik

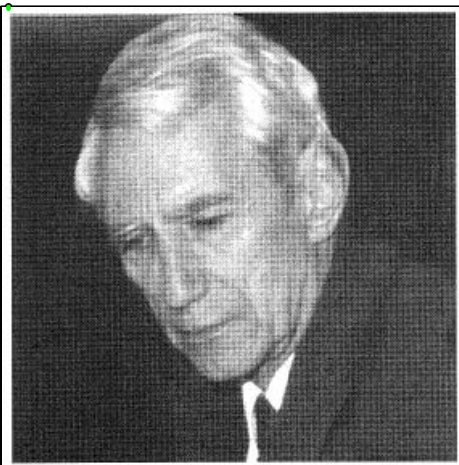
Schach



Verzweigungsfaktor ca. 35-40

Halbzüge	#Stellungen
1	40
2	1600
⋮	
7	164 Milliarden
⋮	
11	419 Billionen ↑

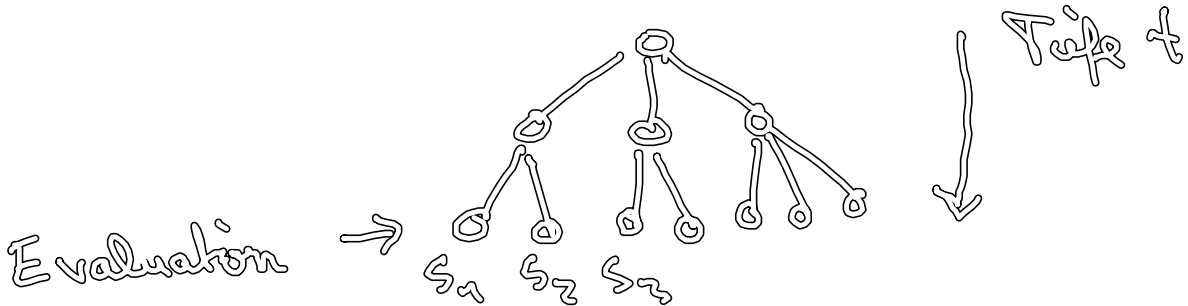
Claude Elwood Shannon



- Mathematiker, New Jersey
- Vortrag, 9. März 49
→ Min Max + Evaluation
- Zufallsmechanische Theorie

IDEA

- feste Suchtiefe
- Abschätzung

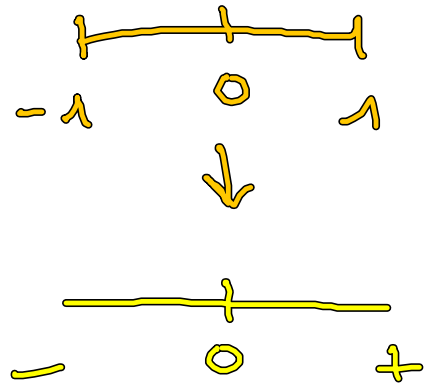


Bewertungsfunktion

meistens linear

S zu beurteilende Stellung

$$f: S \rightarrow \mathbb{R}$$

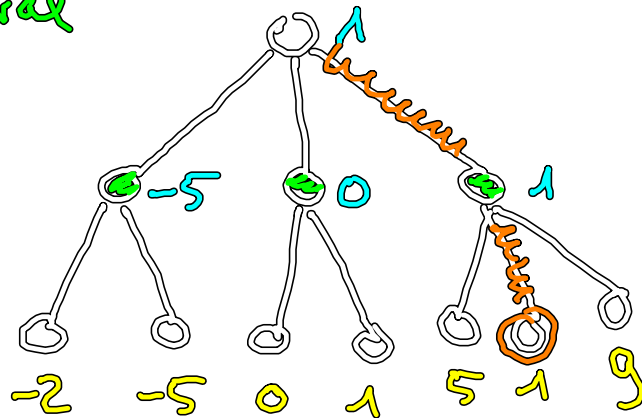


$$f(s) = \sum_{i=1}^n \alpha_i \cdot f_i(s)$$

Spiel-würsen

$$\left\{ \begin{array}{l} f_1(s) = \text{material}(W_s) - \text{material}(S_s) \\ f_2(s) = \text{mobilität}(W_s) - \text{material}(S_s) \\ \vdots \end{array} \right.$$

Min Max + Eval



PseudoCode

max Knoten (Knoten x) → Integer
 if (x ist Blatt)

```

return evaluation(x)
else {
  //  $x_1 \dots x_k$  seien Kinder von  $x$ 
   $\omega := -\infty$ 
  for  $i := 1$  to  $k$  do {
     $v := \text{minKnoten}(x_i)$ 
    if ( $v > \omega$ ) then  $\omega := v$ 
  }
  return  $\omega$ 
}

```

$\text{minKnoten}(\text{Knoten } x) \rightarrow \text{ZuInteger}$
 if (x Blatt)

```

return evaluation(x)
else {
   $\omega := +\infty$ 
  for  $i := 1$  to  $k$  do {
     $v := \text{maxKnoten}(x_i)$ 
    if ( $v < \omega$ ) then  $\omega := v$ 
  }
  return  $\omega$ 
}

```

Min Max in Prolog

minmax (Pos, BestSucc, Val) :-
 moves (Pos, PosList), !,
 best (PosList, BestSucc, Val);
 evaluation (Pos, Val).

best ([Pos], Pos, Val) :-
 minmax (Pos, _, Val).

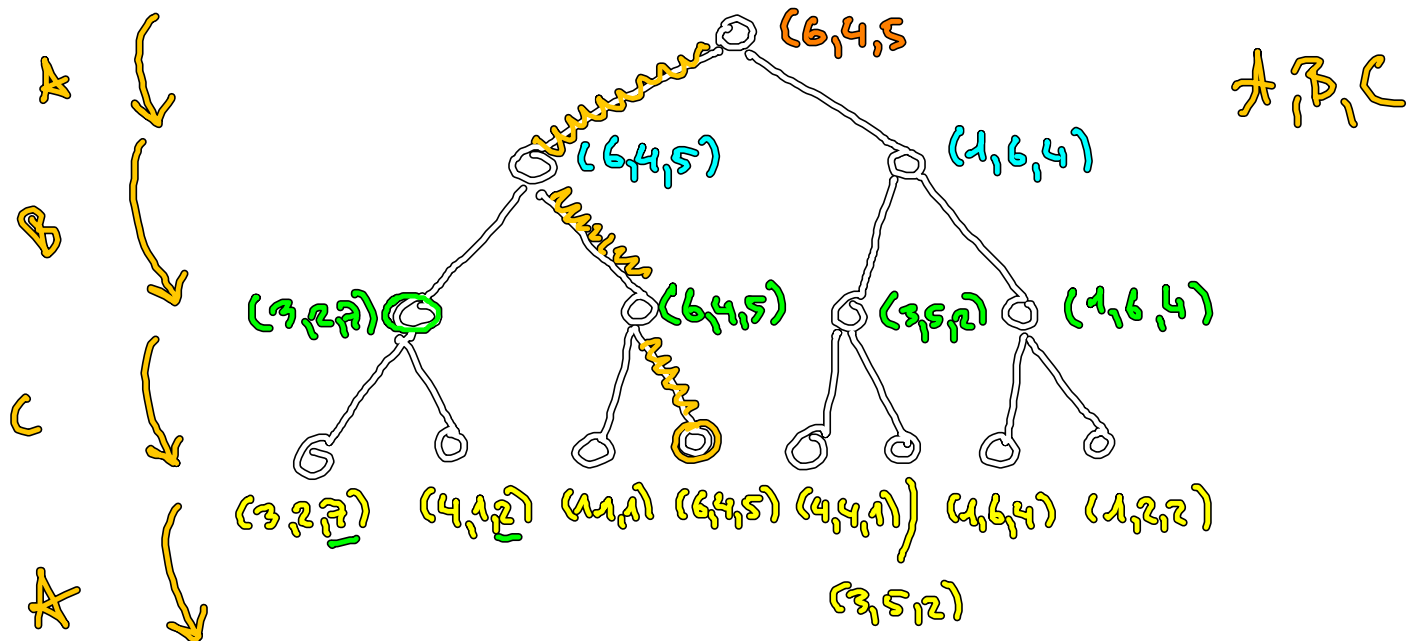
best ([Pos1 | PosList], BestPos, BestVal) :-
 minmax (Pos1, _, Val1),
 best (PosList, Pos2, Val2),
 betterof (Pos1, Val1, Pos2, Val2,
 BestPos, BestVal).

betterof (Pos0, Val0, Pos1, Val1, Pos0, Val0) :-
 min_to_move (Pos0),
 Val0 > Val1, !,
 ;
 max_to_move (Pos0),
 Val0 < Val1, !.

betterof (Pos0, Val0, Pos1, Val1, Pos2, Val2)

Ausblick

Minimax bei Mehr-Spieler-Spielen



(X, 4, 2)
↑ ↑ ↑
A B C

z.B. Siedler

→ Allianzen (Zusammenschlüsse)

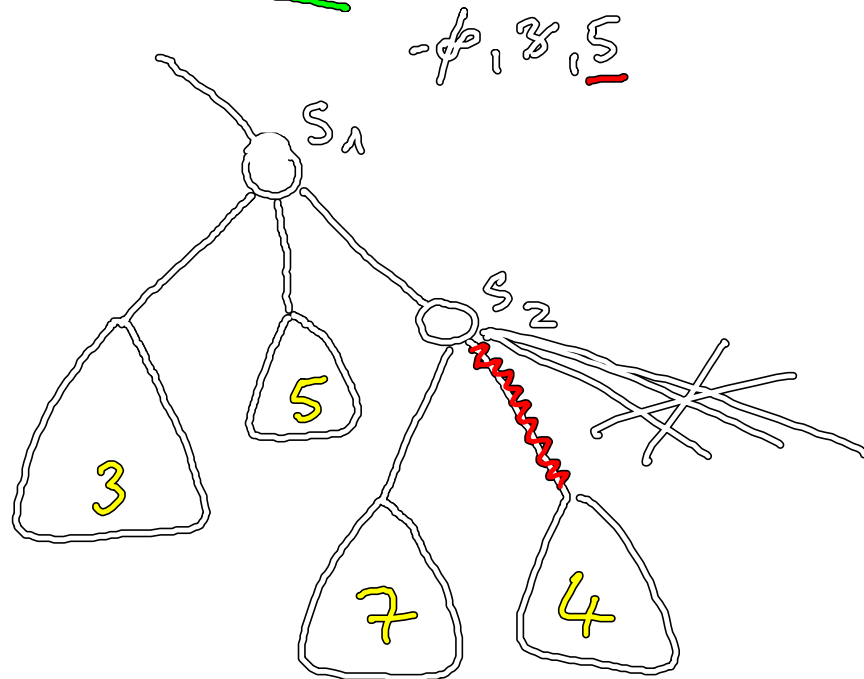
Alan Newell, John Shaw und Herbert Simon

- „Computerwissenschaftler“, Pittsburgh

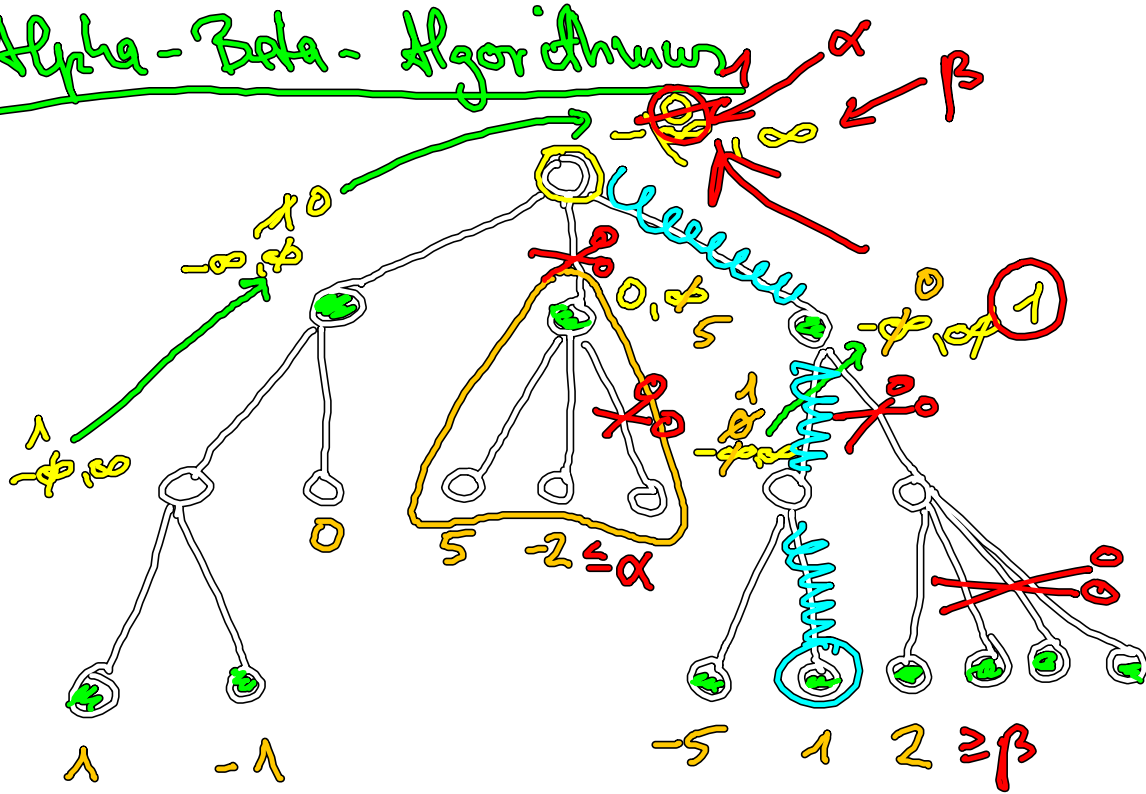
- 1958

→ Alpha-Beta

lokale Betrachtung



Alpha-Beta-Algorithmus



Pseudocode

$\alpha\beta$ max Knoten (Knoten x , Integer α, β) \rightarrow Integer
 if (x ist Blatt)

 return evaluation (x)

else {

$w := \alpha$

 for $i := 1$ to k do {

$v := \alpha\beta$ min Knoten (x_i, w, β)

 if ($v > w$) then $w := v$

 if ($w \geq \beta$) then return w // β -Cuttoff

 }

 return w


```

}
αβ min Knoten (Knoten x, Integer α, β) → Integer
if (x ist Blatt)
    return evaluate(x)
else {
    ω := β
    for i := 1 to k do {
        v := αβ max Knoten(xi, α, ω)
        if (v < ω) then ω := v
        if (ω ∈ α) then return ω
            // α-Cutoff
    }
    return ω
}
}

```

Effektivität von Alpha-Beta

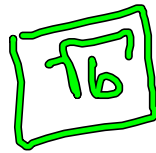
im schlechtesten Fall wie Min Max

Zugreihenfolge

optimal

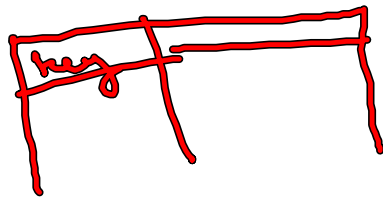
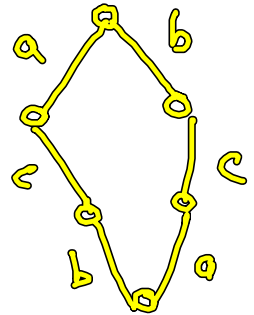
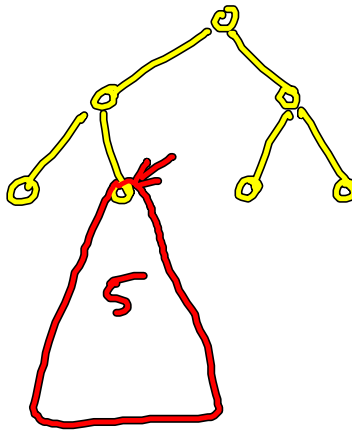
Bsp Schach

Verzweigungsgrad
(branchfaktor b)



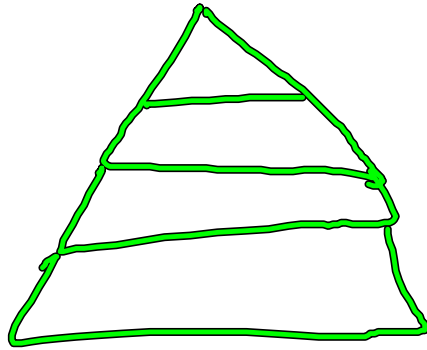
Optimierung

o Transpositionstabellen



o ZugSortierung

Sprach: Killer moves
History move
Hauptvariante



↓ 2, 3, 4

iteratives Suchverfahren