

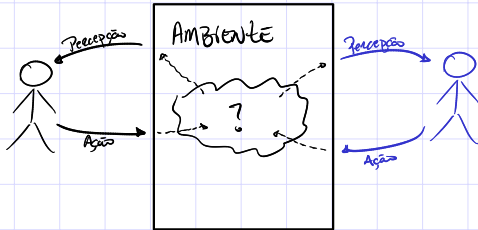
JOGOS COMPETITIVOS

Agentes que competem entre si

AI/MA

CAP. 6

(Jogos Adv.)



→ Objetivos conflitantes?
↳ Busca competitiva

→ Estudo dessas interações

Teoria de Jogos: "Um jogo é uma formulação matemática de interações estratégicas entre agentes independentes que buscam seus próprios interesses."

- ↳ Jogos de tabuleiro / cartas
- ↳ Jogos digitais (videogames)
- ↳ Tráfego
- ↳ Mercado de ações
- ↳ Eleições
- ↳ Geopolítica

- ↳ John Nash (equilíbrio, etc.)
- ↳ Filme "A Beautiful Mind"

Jogos de Soma Zero

- ↳ Competitivo → Quando um ganha: $+1$
↳ O outro perde: -1

Agentes Especializados

- 1958: NSS (Máquina)
- 1997: Deep Blue (Kasparov) - Minimax
- 2016: Alpha Go - MCTS + ML

Algoritmo Minimax

- ↳ 2 AGENTES
- ↳ Completamente observável
- ↳ Determinístico
- ↳ Alternância de decisão

XADREZ, DAMAS, VELHA,
OTHERO/REVERSI

ELEMENTOS

- ↳ Estados
- ↳ Ações
- ↳ Função Sucessor

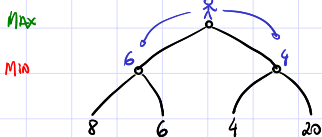
UTILIDADE → Valor de um estado (+1, -1)

Agentes → MAX (Jogador)

↳ MIN (Oponente)

} Ponto de vista do agente/jogador

↳ Regra Geral: Escolher a melhor jogada possível, considerando que o adversário fará o mesmo!



} MAX: Maximiza a utilidade
MIN: Minimiza a utilidade

MINIMAX (s):

$V \leftarrow \text{VALOR-MAX}(s)$

Retorna Ação cujo sucessor tem valor V

VALOR-MAX(s):

Se s é terminal: Retorna $U(s)$

$V \leftarrow -\infty$

PARA CADA s' SUCESSOR DE s :

$V \leftarrow \text{MAX}(V, \text{VALOR-MIN}(s'))$

Retorna V

VALOR-MIN(s):

Se s é terminal: Retorna $U(s)$

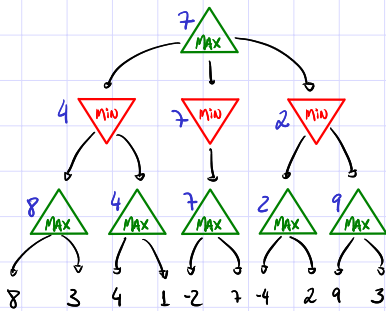
$V \leftarrow +\infty$

PARA CADA s' SUCESSOR DE s :

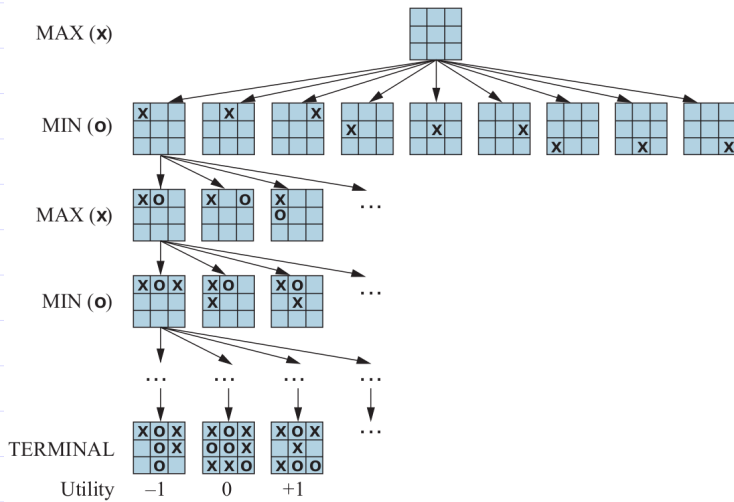
$V \leftarrow \text{MIN}(V, \text{VALOR-MAX}(s'))$

Retorna V

EXEMPLO



EXEMPLO - JOGO DA VELHA



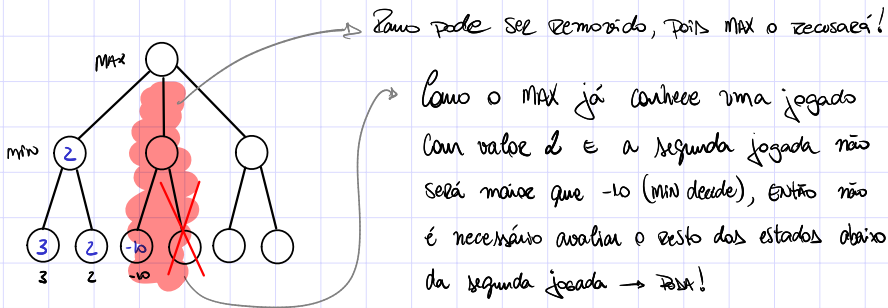
PODA ALFA-BETA

Minimax = Força Bruta (Enviável p/ jogos complexos)

↳ XADREZ: 35 jogadas em 100 turnos $\approx 10^{154}$ estados
↳ GO: 250 jogadas em 150 turnos $\approx 10^{359}$ estados

↳ 1 segundo = 10^9 estados avaliados,
Decisão tomada em $\begin{cases} 10^{145} \text{ segundos (XADREZ)} \\ 10^{350} \text{ segundos (GO)} \end{cases}$ $\left\{ \begin{array}{l} \text{Idade do Universo} \approx 10^{12} \text{ segundos!} \end{array} \right.$

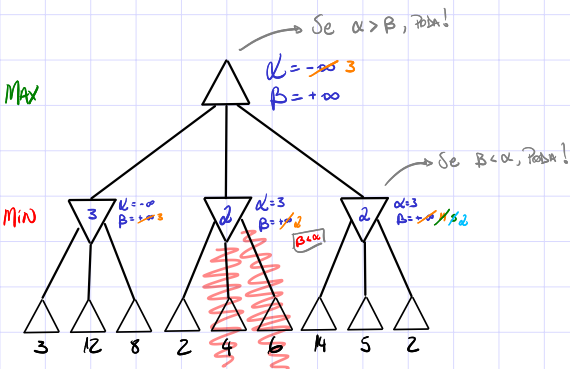
PODA NA PODA ALFA-BETA



Implementação

α = Melhor escolha conhecida para MAX (maior valor)

β = Melhor escolha conhecida para MIN (menor valor)



Importante: Poda Alfa-Beta mantém a otimalidade do Minimax!

Algoritmo (MAXMIN com Bônus ALFA-BETA)

MINIMAX (s):

$V \leftarrow \text{VALOR-MAX}(s, -\infty, +\infty)$

Retorna Ação cujo sucessor tem valor V

VALOR-MAX (s, α , β)

Se s é terminal: Retorna $U(s)$

PARA CADA s' SUCESSOR DE s :

$V \leftarrow \text{VALOR-MIN}(s', \alpha, \beta)$

$\alpha \leftarrow \text{MAX}(\alpha, V)$

SE $\beta \leq \alpha$: Retorna α

Retorna α

VALOR-MIN (s, α , β):

Se s é terminal: Retorna $U(s)$

PARA CADA s' SUCESSOR DE s :

$V \leftarrow \text{VALOR-MAX}(s', \alpha, \beta)$

$\beta \leftarrow \text{MIN}(\beta, V)$

SE $\alpha > \beta$: Retorna β

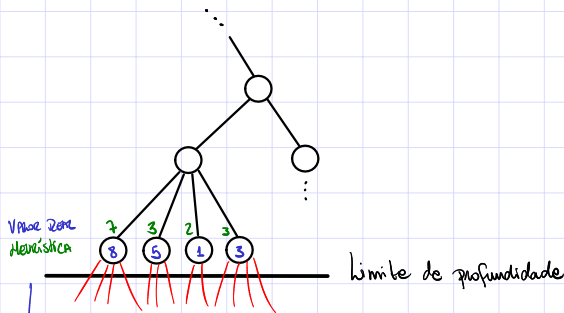
Retorna β

Minimax com Profundidade Limitada (Alpha-Beta Heurístico)

- ↳ Mesmo com a poda, a árvore completa é considerada. → Evitar em alguns casos!
- ↳ Solução: limitar a profundidade da busca!

Definição: Para a busca em uma dada profundidade; aplicar uma heurística para avaliar estados de parada não terminais!

Heurística: função que estima a utilidade de um estado;
ou seja, estima o resultado do minimax a partir do estado!



Boa heurística, pois diferencia os estados e identifica corretamente o melhor (max ou min)!

Como modelar Heurísticas?



Para cada possibilidade, avaliar fatores como:

- Peças capturadas → Valor material (Rei = 1, ...)
- Cheque
- Ameaça
- Mobilidade
- Etc...

↳ Cada fator f_i recebe um peso w_i

$$AVALIA(s) = \sum_{i \in [n]} w_i f_i(s)$$

Outras MEMÓRIAS

- Busca por Aprofundamento iterativo
(enquanto há tempo, avalia mais um nível da árvore)
- Busca Quiescente (estado quiescente = onde a heurística não varia)
(se heurística varia muito, expande mais um nível)
- Extensões singulares
(Pelo conhecimento do domínio, sabe-se que uma jogada é muito melhor que as outras → escolhe ela e ignora as demais)
↳ ex: captura da rainha!
- Poda adiantada
(Similar ao anterior; Sabendo que uma jogada é muito ruim, poda o ramo da árvore)
- Bancos de dados de jogadas

Outro Algoritmo? Monte Carlo Tree Search (MCTS)!