

procura adversarial imperfeita

viabilidade do minimax-alfabeta

mesmo com estados ordenados por “melhores 1º”
a complexidade temporal é

$$O(b^{m/2})$$

o que só é viável em jogos muito simples

em jogos mais elaborados é inviável pesquisar tantos nós

ex: xadrez $> O(10^{40})$

procura adversarial limitada

cortar a procura a uma profundidade limitada:

- estimar o custo do ponto de corte com uma função de avaliação heurística E_{VAL}
- substituir o teste terminal por um teste de corte $CUTOFF-TEST$

(ideia proposta por Shannon em 1950)

minimax heurístico

- recebe o estado atual s e a profundidade máxima de procura d

$$\text{H-MINIMAX}(s, d) = \begin{cases} \text{EVAL}(s) & \text{se CUTOFF-TEST}(s, d) \\ \max_{a \in \text{ACTIONS}(s)} \text{H-MINIMAX}(\text{RESULT}(s, a), d+1) & \text{se JOGADOR}(s) = \text{MAX} \\ \min_{a \in \text{ACTIONS}(s)} \text{H-MINIMAX}(\text{RESULT}(s, a), d+1) & \text{se JOGADOR}(s) = \text{MIN} \end{cases}$$

um caso

suponhamos um computador que consegue explorar

10 k nós / s

e um jogo que tem um limite de

100 s / jogada

então o computador consegue explorar

1 M nós / jogada

no minimax-alfabeta atinge-se profundidade ~ 8 no xadrez

que é um programa jogador razoável

h-minimax

perde-se a otimalidade

a qualidade do jogador depende fortemente da qualidade da heurística

uma pequena variação na profundidade pode fazer uma grande diferença na qualidade do jogador

usar aprofundamento progressivo para um algoritmo “em qualquer altura” (*anytime*) ↔ jogadas em tempo-real

vai explorando até ter de produzir uma resposta (limite de tempo)

função de avaliação

é fundamental manter a ordem dos estados

estimar com mais valor estados que levem a vitória, ...

modelo típico: soma ponderada de avaliação de características do estado

$$Eval(s) = w_1 f_1(s) + w_2 f_2(s) + \dots + w_n f_n(s) = \sum_{i=1}^n w_i f_i(s)$$

ex. xadrez: valor das peças (w); estrutura de peões; segurança do rei (traduzidos em unidade “valor de um peão”)

adaptando o alfabeto para h-

em vez de

if `TERMINAL-TEST` (*state*) **then return** `UTILITY` (*state*)

usa-se

if `CUTOFF-TEST`(*state*, *depth*) **then return** `EVAL` (*state*)

tem de retornar *V* também nos estados terminais

aprofundamento progressivo “resolve” problema da ordem das ações

peões

minimaxcom profundidade 3

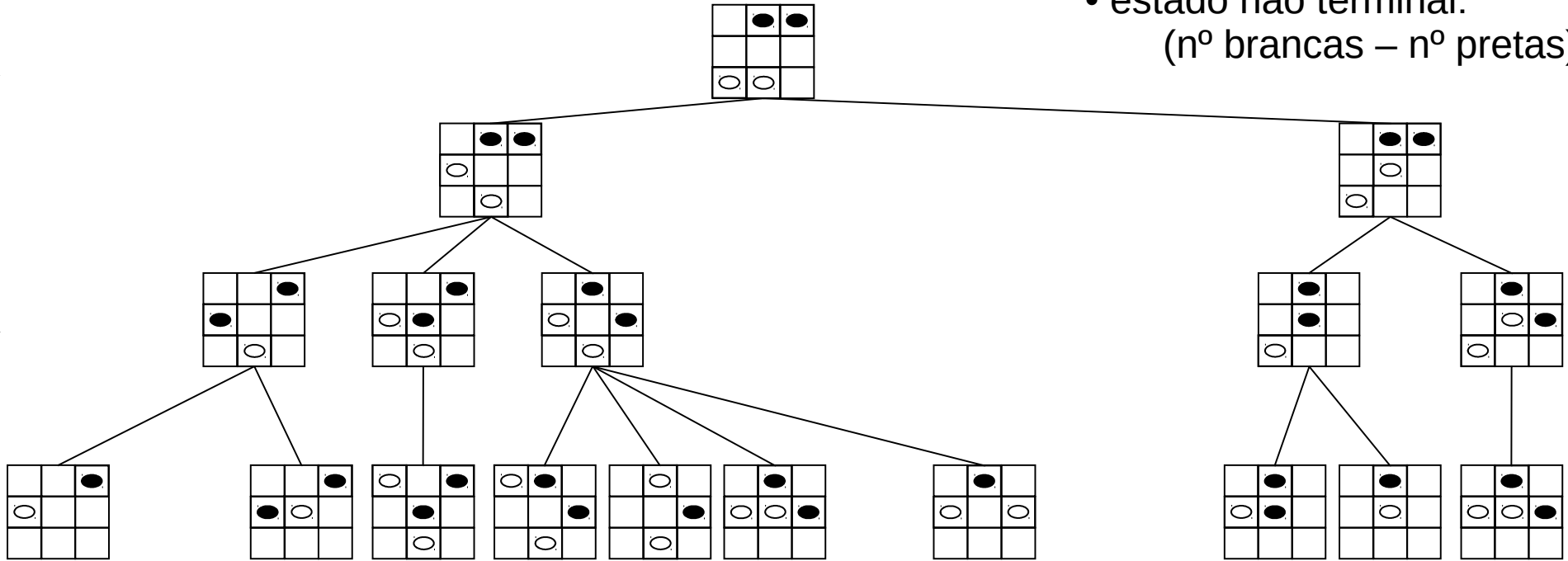
função de avaliação

- vitória das brancas: 2
- vitória das pretas: -2
- empate: 0
- estado não terminal:
(nº brancas – nº pretas)

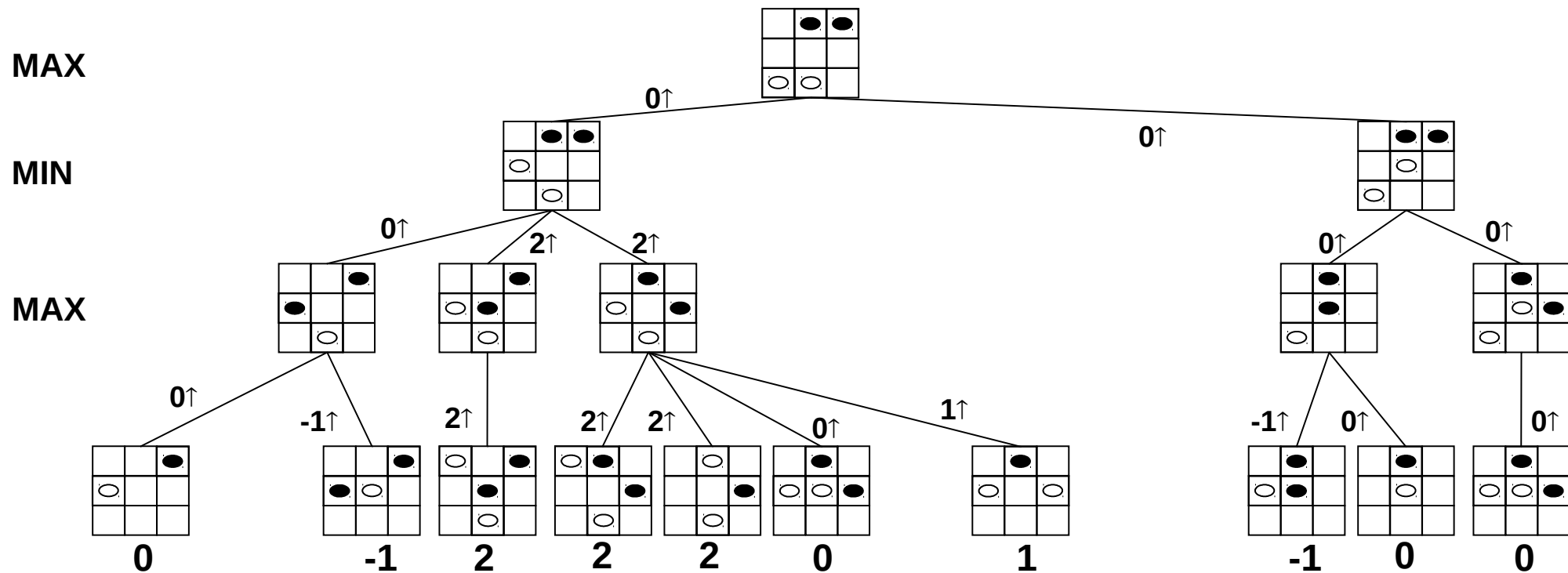
MAX

MIN

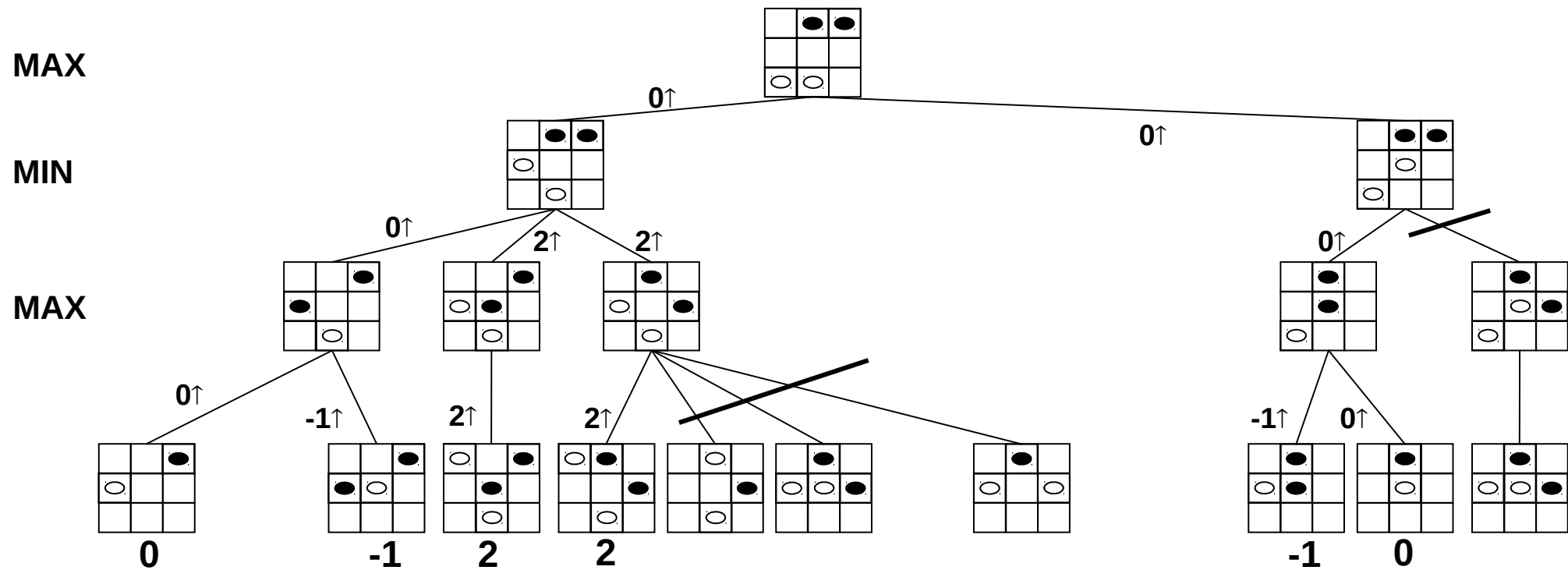
MAX



valores do minimax



alfa-beta



e ainda

procura em tabelas

por estado

guarda a melhor ação nesse estado

ex: aberturas, e finais de jogo em xadrez

aprendizagem automática

para preencher as tabelas

em alternativa ao conhecimento de peritos

jogos estocásticos

jogadas envolvem algum grau de aleatoriedade

é necessário incluir **nós de acaso**

para além dos nós MAX e MIN



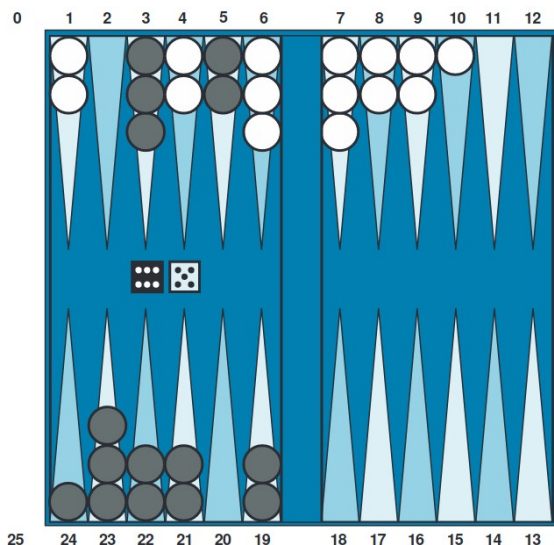
expectiminimax

nós passam a ter **valor esperado** (em vez de definido)

nós MAX e MIN mantêm valores como no minimax

nós de acaso: média ponderada dos descendentes

gamão



fonte: AIMA 4ª ed.

BKG 9.8 - 1º jogador
IA a bater um
campeão mundial
em jogos (1979)

MAX

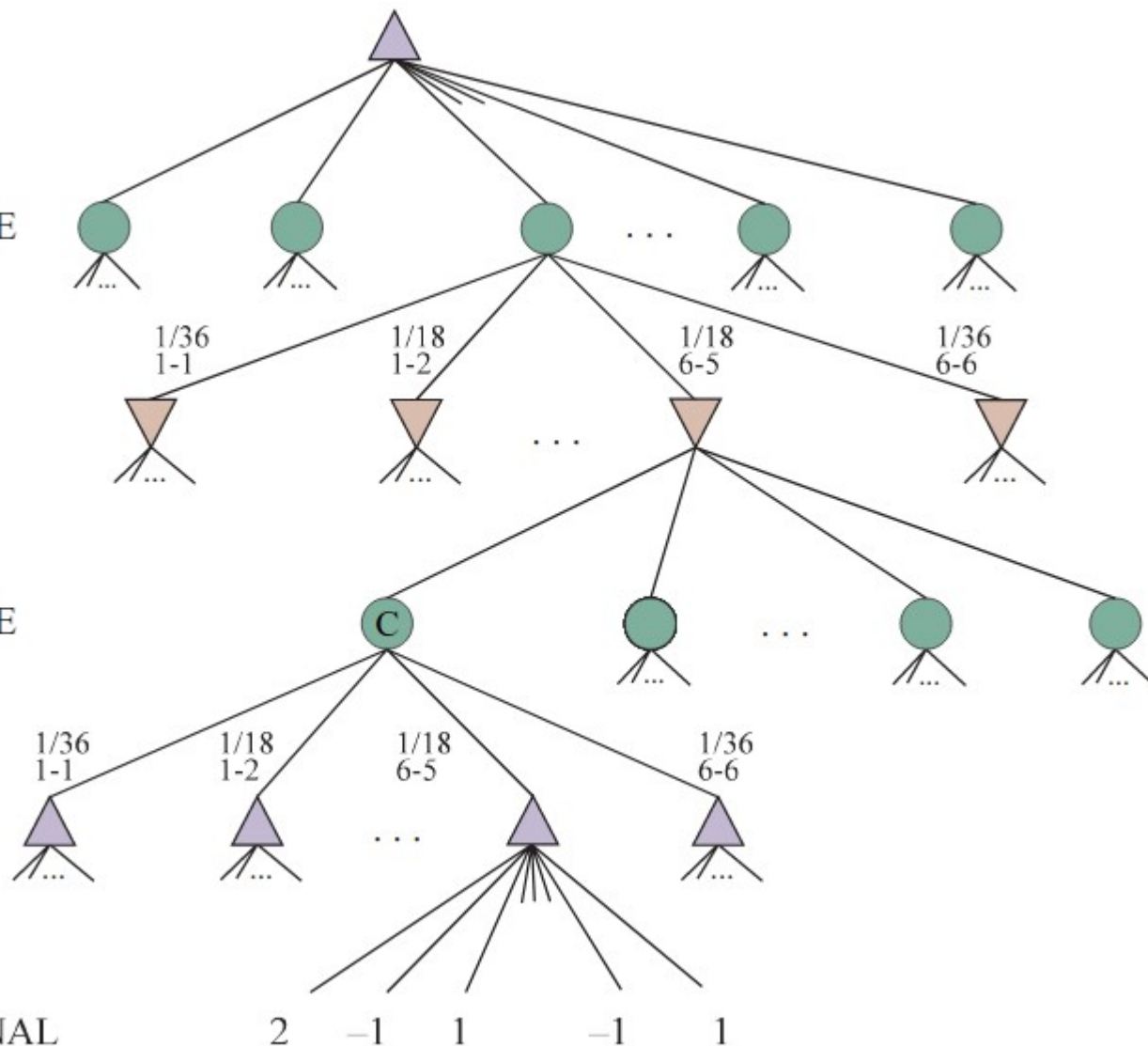
CHANCE

MIN

CHANCE

MAX

TERMINAL



nos nós de acaso

adaptação do minimax

EXPECTIMINIMAX(s) =

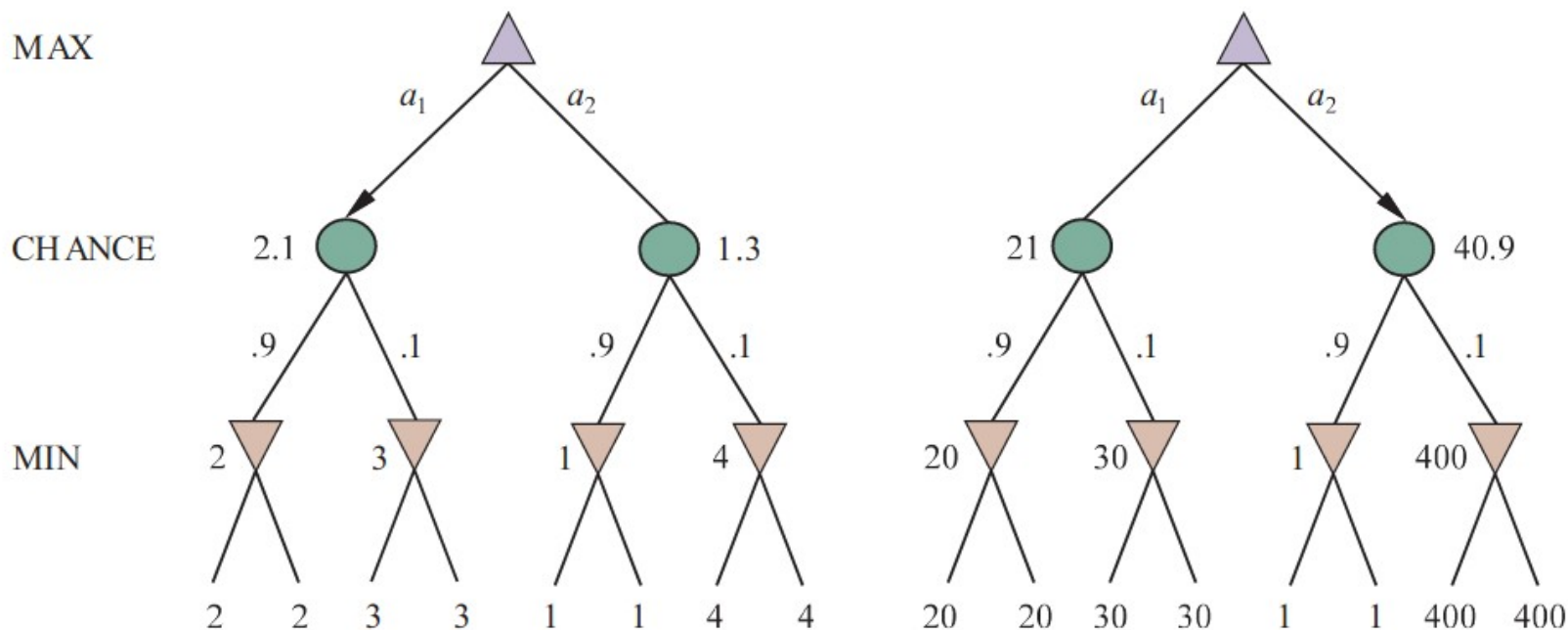
$$\left\{ \begin{array}{ll} \text{UTILITY}(s) & \text{se } \text{TERMINAL-TEST}(s) \\ \max_{a \in \text{ACTIONS}(s)} \text{EXPECTIMINIMAX}(\text{RESULT}(s, a)) & \text{se } \text{JOGADOR}(s) = \text{MAX} \\ \min_{a \in \text{ACTIONS}(s)} \text{EXPECTIMINIMAX}(\text{RESULT}(s, a)) & \text{se } \text{JOGADOR}(s) = \text{MIN} \\ \sum_r P(r) \text{EXPECTIMINIMAX}(\text{RESULT}(s, r)) & \text{se } \text{JOGADOR}(s) = \text{ACASO} \end{array} \right.$$

r – resultado do evento de acaso

função de avaliação

tem de ser uma transformação linear da probabilidade de vencer numa posição

senão...



uma aproximação

método de simulação Monte-Carlo

- fazer o algoritmo jogar milhares de jogos contra ele próprio usando resultados do acaso gerados aleatoriamente

- obter as frequências de vitória das diversas ações

 - considerar essas frequências a probabilidade do resultado no estado (em jogos de dados diz-se *rollout*)

atualmente, aprendizagem por reforço profunda dá muito melhores resultados

outro tipo de jogos

jogos parcialmente observáveis

em que não se conhece completamente o estado do jogo

ex: bridge, king,... (aleatoriedade inicial e depois informação incompleta)

etc.

importância do meta-raciocínio – raciocínio acerca de valer ou não a pena raciocinar mais em certos estados/ramos do jogo